



①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ Übersetzung der  
europäischen Patentschrift

⑧7 EP 0 153 877 B1

⑩ DE 35 87 375 T 2

⑤1 Int. Cl. 5:-  
G 06 F 3/12  
G 06 F 15/64  
G 06 F 5/06

②1	Deutsches Aktenzeichen:	35 87 375.2
⑧6	Europäisches Aktenzeichen:	85 301 405.8
⑧6	Europäischer Anmeldetag:	28. 2. 85
⑧7	Erstveröffentlichung durch das EPA:	4. 9. 85
⑧7	Veröffentlichungstag der Patenterteilung beim EPA:	2. 6. 93
④7	Veröffentlichungstag im Patentblatt:	23. 9. 93

DE 35 87 375 T 2

③0 Unionspriorität: ③2 ③3 ③1

29.02.84 JP 37883/84

⑦3 Patentinhaber:

Fujitsu Ltd., Kawasaki, Kanagawa, JP

⑦4 Vertreter:

Seeger, W., Dipl.-Phys.; Seeger, A., Dipl.-Phys.,  
Pat.-Anwälte, 81369 München

⑧4 Benannte Vertragsstaaten:

BE, DE, FR, GB, IT, NL

⑦2 Erfinder:

Itoh, Sumio, Kawasaki-shi Kanagawa 213, JP

⑤4 Schaltung zur Zwischenspeicherung von Bilddaten.

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 35 87 375 T 2

Die vorliegende Erfindung bezieht sich auf eine Bilddatenpufferschaltungsanordnung.

Ein Laserdrucker, zum Beispiel, soll im allgemeinen Bilddaten ständig mit hoher Geschwindigkeit ausdrucken, und demzufolge ist eine Hochgeschwindigkeitsübertragung von Bilddaten von einem Bildspeicher zu dem Laserdrucker notwendig.

Um ständig Bilddaten zu lesen, können in einer Bilddatenpufferschaltungsanordnung zwei Bildspeicher vorgesehen sein, so daß, während einer der Bildspeicher eine Seite von Bilddaten speichert, eine andere Seite von Bilddaten aus dem anderen Bildspeicher gelesen werden kann. Dieser Aufbau erfordert jedoch zwei Bildspeicher zum gleichzeitigen Speichern bzw. Lesen einer Seite von Bilddaten, mit dem Resultat, daß die Speicherkapazität insgesamt groß sein muß.

Es ist möglich, bei einer Bilddatenpufferschaltungsanordnung einen Bildspeicher zu verwenden, der eine Kapazität zum Speichern von gerade einer Seite von Bilddaten hat, bei dem neue Bilddaten sequentiell in einen Bereich geschrieben werden können, aus dem die vorherigen Bilddaten gelesen worden sind. Solch ein Bildspeicher kann jedoch das Lesen mit hoher Geschwindigkeit nicht erleichtern, wenn die geschriebenen Bilddaten zufällig geschrieben werden, d. h., nicht in Reihenfolge von oben nach unten der Seite geschrieben werden, wie später unter Bezugnahme auf Zeichnungen ausführlicher beschrieben.

Gemäß der vorliegenden Erfindung ist eine Bilddatenpufferschaltungsanordnung vorgesehen mit:

einem Speicher, der eine Vielzahl von individuell adressierbaren Datenspeicherplätzen hat, zum Speichern von Daten, die ein Bild definieren, das in eine Vielzahl von räumlichen Einheiten unterteilt ist;

einem Adreßspeichermittel zum Speichern von Adressen von solchen Plätzen, die zum Speichern von neuen Daten

verfügbar sind;

einem Adreßkorrespondenzspeicher, der Speicherbereiche hat, die jeweilig den genannten räumlichen Einheiten entsprechen und mittels entsprechender Adreßkodes zugänglich sind;

5 einem Schreibmittel zum Schreiben der bilddefinierenden Daten, falls vorhanden, von irgendeiner ausgewählten der genannten räumlichen Einheiten in einen individuellen von solchen verfügbaren Plätzen, welcher verfügbarer Platz durch  
10 Entnahme der Adresse jenes Platzes aus dem Adreßspeichermittel ausgewählt worden ist, und zum Ablegen der Adresse jenes ausgewählten Platzes in dem individuellen Speicherbereich des Korrespondenzspeichers, der der gewählten räumlichen Einheit entspricht; und

15 einem Lesemittel zum Verwenden des Adreßkodes, der für den Speicherbereich passend ist, der einer ausgewählten der genannten räumlichen Einheiten entspricht, um aus dem genannten Speicher die bilddefinierenden Daten, falls vorhanden, der ausgewählten räumlichen Einheit auszulesen,  
20 und zum Wiederhinzufügen der Adresse des Speicherplatzes, aus dem jene bilddefinierenden Daten ausgelesen worden sind, zu dem genannten Adreßspeichermittel.

Vorzugsweise hat die Schaltungsanordnung zwei solcher Adreßkorrespondenzspeicher zur jeweiligen Verwendung beim  
25 Lesen/Schreiben von bilddefinierenden Daten von zwei verschiedenen Bildern, wobei die Operation der Schreib- und Lesemittel so ist, daß Daten, die eines der Bilder definieren, in den genannten Speicher geschrieben werden können, während Daten, die das andere Bild definieren, aus jenem  
30 Speicher gelesen werden.

Die Anzahl der genannten Speicherbereiche in dem genannten Adreßkorrespondenzspeicher kann wenigstens das Zweifache der Anzahl der genannten Datenspeicherplätze in dem genannten Speicher betragen.

Jeder der genannten Speicherbereiche in dem Korrespondenzspeicher kann ein ihm zugeordnetes individuelles korrespondenzanzeigendes Bit haben, zum Anzeigen, ob bilddefinierende Daten der entsprechenden räumlichen Einheit vorhanden sind oder nicht, die in dem Speicher an einem Platz gespeichert sind, dessen Adresse in dem betreffenden Speicherbereich gehalten ist, wobei die Schaltungsanordnung so ist, daß das genannte korrespondenzanzeigende Bit, das dem entsprechenden Speicherbereich zugeordnet ist, durch das Schreibmittel gesetzt wird, wenn die bilddefinierenden Daten der ausgewählten räumlichen Einheit in den Speicher geschrieben sind, und daß das genannte korrespondenzanzeigende Bit, das dem entsprechenden Speicherbereich zugeordnet ist, durch das Lesemittel zurückgesetzt wird, wenn die bilddefinierenden Daten jener räumlichen Einheit aus dem Speicher gelesen werden.

Bei einer Ausführungsform der vorliegenden Erfindung erzeugt das genannte Lesemittel beim Feststellen, daß das genannte korrespondenzanzeigende Bit, das einem Speicherbereich zugeordnet ist, der zum Lesen ausgewählt ist, in dem Rücksetzzustand ist, ein Leerdatensignal als bilddefinierende Daten für die entsprechende räumliche Einheit.

Das genannte Adreßspeichermittel kann eine First-in-First-out-Schaltung sein.

Der genannte Korrespondenzspeicher kann auch individuelle korrespondenzsperrende Bits haben, die jeweilig den genannten Datenspeicherbereichen zugeordnet sind, welche Bits gesetzt werden können, um zu verhindern, daß die bilddefinierenden Daten der entsprechenden räumlichen Einheiten in den genannten Speicher geschrieben werden.

Bei einer Ausführungsform der vorliegenden Erfindung erzeugt das genannte Lesemittel beim Feststellen, daß das korrespondenzsperrende Bit, das einem Speicherbereich zugeordnet ist, der zum Lesen ausgewählt ist, gesetzt ist,

ein Leerdatensignal als bilddefinierende Daten für die entsprechende räumliche Einheit.

Das Schreibmittel und das Lesemittel umfassen vorzugsweise ferner ein Adressenkonvertierungsmittel zum Konvertieren der Schreibadresse oder der Leseadresse in eine seitlich zugängliche Adresse oder eine vertikal zugängliche Adresse.

Eine Bilddatenpufferschaltungsanordnung, die die Erfindung verkörpert, kann so konstruiert sein, daß sie zu einer Hochgeschwindigkeitsverarbeitung in der Lage ist, und kann ein verbessertes virtuelles Speichersystem vorsehen, das geeignet ist, um Daten von einem Speicher an einen Drucker, wie einen Laserdrucker, auszugeben.

An Hand eines Beispiels wird nun auf die beiliegenden Zeichnungen Bezug genommen, in denen:

Fig. 1 ein Blockdiagramm eines Datenverarbeitungssystems ist;

Fig. 2 ein Blockdiagramm eines Beispiels einer Bilddatenpufferschaltungsanordnung zeigt, die die vorliegende Erfindung nicht verkörpert;

Fig. 3A ein schematisches Diagramm eines Bildspeichers bei einem anderen Beispiel einer Bilddatenpufferschaltungsanordnung zeigt, die die vorliegende Erfindung nicht verkörpert;

Fig. 3B eine schematische Ansicht zum Darstellen einer Datenspeicherung in einem logischen Speicherraum des Bildspeichers von Fig. 3A zeigt;

Fig. 4 ein Blockdiagramm einer Bilddatenverarbeitungsanordnung ist, die eine Bilddatenpufferschaltungsanordnung enthält, die die vorliegende Erfindung verkörpert;

Fig. 5 ein Blockschaltungsdiagramm einer Bilddatenpufferschaltungsanordnung ist, die die vorliegende Erfindung verkörpert;

Fig. 6 Zustände einer Zellenadreß-Umsetzungstabelle (CAT), eines ZellenSpeichers (CM) und einer Zellenadreß-

Warteschlange (CAQ) zeigt, die in der Schaltungsanordnung von Fig. 5 enthalten sind, und auch die Datenspeicherung in einem logischen Speicher zeigt, der innerhalb des Zellen-speichers (CM) definiert ist;

5        Fig. 7A Anfangszustände der zwei CATs und der CAQ zeigt;

      Fig. 7B Übergangszustände der zwei CATs, der CAQ und des logischen Speichers zeigt, wenn neue Daten in eine Seite des logischen Speichers geschrieben werden;

10       Fig. 7C Fig. 7B entspricht, aber solche Zustände darstellt, wenn weitere neue Daten in eine andere Seite des logischen Speichers geschrieben werden;

      Fig. 8 ein Flußdiagramm zum Erläutern der Operation der in Fig. 5 gezeigten Schaltungsanordnung ist;

15       Fig. 9 ein Schaltungsdiagramm einer Bildspeicher-steuereinheit in der in Fig. 5 gezeigten Schaltungsanordnung ist;

      Fig. 10 die Struktur der Zellenadreß-Umsetzungstabelle (CAT) in der Schaltungsanordnung von Fig. 5 zeigt;

20       Fig. 11 die Struktur des ZellenSpeichers (CM) in der Schaltungsanordnung von Fig. 5 zeigt;

      Fig. 12 ein Blockdiagramm von weiteren Teilen der Schaltungsanordnung von Fig. 5 ist; zum Erläutern derer vertikaler und horizontaler Zugriffsmodi;

25       Fig. 13A und 13B entsprechende schematische Ansichten zeigen, die eine Korrespondenz zwischen Plätzen in dem obengenannten logischen Speicher und Positionen auf dem Druckpapier darstellen;

      Fig. 14A und 14B entsprechende Ansichten zeigen, die die Anordnung von gedruckten Zeichen auf den in den Figuren 13A bzw. 13B gezeigten Druckpapieren darstellen;

30       Fig. 15A ein Flußdiagramm zum Erläutern eines Datenstromes von einem Kanalcontroller (CHC) zu der Bildhardware (IMH) in der in Fig. 4 gezeigten Bilddatenverarbeitungsan-

ordnung ist;

Fig. 15B ein Datenformat eines Anweisungskodes zeigt, der von einer zentralen Verarbeitungseinheit (CPU), die in Fig. 1 gezeigt ist, zu einem Steuerprozessor (CTP), der in Fig. 4 gezeigt ist, übertragen wird;

Fig. 15C ein Datenformat bei einem in Fig. 4 gezeigten Formularsteuerungspuffer (FCB) zeigt;

Fig. 15D ein Verfahren des Umsetzens eines Zeichenkodes in die entsprechende obere Adresse des Zeichens zeigt;

Fig. 15E ein Datenformat bei einem in Fig. 4 gezeigten Seitenpuffer (PB) zeigt;

Fig. 15F eine Startposition von jedem Zeichen graphisch darstellt;

Fig. 16 ein Blockdiagramm ist, das einen in Fig. 4 gezeigten Zeichengeneratorcontroller (CGC) ausführlich zeigt;

Fig. 17 ein Blockdiagramm ist, das eine in Fig. 16 gezeigte Schreibschaltung (WC) ausführlich zeigt;

Fig. 18A ein Flußdiagramm zum Erläutern einer Schreibsteueroperation ist;

Fig. 18B ein Diagramm zum Erläutern eines seitlichen Zugriffs und eines vertikalen Zugriffs bei der in Fig. 18A gezeigten Schreibsteueroperation ist;

Fig. 19 ein Blockdiagramm eines Druckcontrollers in der Anordnung von Fig. 4 ist;

Fig. 20 ein ausführliches Blockdiagramm eines Teils des in Fig. 19 gezeigten Druckcontrollers ist;

Fig. 21A ein ausführliches Blockdiagramm eines anderen Teils des in Fig. 19 gezeigten Druckcontrollers ist; und

Fig. 21B ein Schaltungsdiagramm einer Takterzeugungsschaltungsanordnung zum Erzeugen von Taktsignalen ist, die von dem Druckcontroller von Fig. 19 verwendet werden.

Figur 1 zeigt ein Datenverarbeitungssystem zum Erläutern der Ausgangssituation der vorliegenden Erfindung.

In Fig. 1 enthält das Datenverarbeitungssystem eine zentrale Host-Verarbeitungseinheit (CPU) 1, einen Hauptkanal 2, eine Bilddatenverarbeitungsanordnung 3 und eine Eingabe-/Ausgabeeinheit (E/A) 4 als Anzeigeeinheit. Die Bilddatenverarbeitungsanordnung 3 enthält eine Bilddatenpufferschaltungsanordnung 30, die zu der vorliegenden Erfindung gehört, und einen Drucker (in Fig. 1 nicht gezeigt). Die CPU 1 kommuniziert durch den Hauptkanal 2 mit der Bilddatenverarbeitungsanordnung 3 oder der E/A-Einheit 4. Zeichendaten von Zeichen, die von der Bilddatenverarbeitungsanordnung 3 auszudrucken sind, oder gewisse Anweisungen zum Betreiben der Bilddatenverarbeitungsanordnung 3 werden von der CPU 1 erzeugt und durch den Kanal 2 zu der Bilddatenverarbeitungsanordnung 3 übertragen.

Die Bilddatenpufferschaltungsanordnung dient als Datenpuffer zum Übertragen einer Seite von Bilddaten, die an eine Druckerbaugruppe (in Fig. 1 nicht gezeigt) auszugeben sind. Die Schaltungsanordnung 30 kann als Abwandlung eines virtuellen Speichersystems angesehen werden.

Übrigens ist ein unterschiedliches Beispiel der Verwendung eines virtuellen Speichersystems in bezug auf Bilddaten in EP-A-0 099 989 gegeben, das eine Bildanzeigesteuervorrichtung offenbart, die einen Bildspeicher zum Speichern von Bilddaten enthält, die (auf einem Anzeigeschirm) anzuzeigen sind. Der Bildspeicher ist auf virtueller Basis organisiert, indem ein Einteilungsspeicher (Adreßkorrespondenzspeicher) verwendet wird, der Speicherbereiche hat, die jeweilig den räumlichen Einheiten (Teilen des Bildes) entsprechen, die anzuzeigen sind, und mittels entsprechender Adreßkodes zugänglich sind. Bei dieser Vorrichtung jedoch macht die Leseoperation den Lesespeicherplatz des Bildspeichers für das Speichern von neuen Bilddaten nicht frei.

Figur 2 zeigt ein Beispiel einer Bilddatenpufferschaltungsanordnung, die die Erfindung nicht verkörpert. In Fig.



2 enthält die Schaltungsanordnung zwei Bildspeicher ( $IMM_1$  und  $IMM_2$ ) 30a1 und 30a2 und einen Selektor (SEL) 30a3. Jeder der zwei Bilddatenspeicher 30a1 und 30a2 hat eine Kapazität zum Speichern einer Seite von Bilddaten. Während des Lesens einer Seite von Bilddaten von dem ersten Bilddatenspeicher 30a1 zu einem Laserdrucker, zum Beispiel, wählt der Selektor den zweiten Bildspeicher 30a2 für eine Schreiboperation aus. Bei der Schreiboperation wird auf den zweiten Bilddatenspeicher 30a2 zum Einschreiben einer Seite von Daten zugegriffen, indem logische Adressen LA spezifiziert werden, die dem logischen Adreßraum des zweiten Bildspeichers 30a2 entsprechen. Ähnlich ist es während der Leseoperation des zweiten Bildspeichers 30a1 zum Schreiben einer nächsten Seite von Bilddaten. Dieser Aufbau gewährleistet eine Leseoperation mit hoher Geschwindigkeit, da die Bilddaten auf Grund einer Schreiboperation ständig ohne Stop gelesen werden können.

Die oben beschriebene Bilddatenpufferschaltungsanordnung benötigt jedoch die zwei Bildspeicher, oder zwei Seiten an Speicherkapazität, was zu dem Nachteil führt, eine große Speicherkapazität zu benötigen.

Um die Speicherkapazität zu reduzieren, kann ein einzelner Bildspeicher mit einer Speicherkapazität von einer Seite vorgesehen werden, wie in Fig. 3A gezeigt. In Fig. 3A stellt ein Teil A, der durch schraffierte Linien gekennzeichnet ist, einen logischen Raum dar, in dem Bilddaten so gelesen worden sind, daß dieser Teil A zum Schreiben von Daten verwendet werden kann, und ein Teil B stellt einen logischen Raum dar, aus dem Bilddaten nicht gelesen worden sind. Falls Schreibdaten sequentiell von oben nach unten des logischen Raumes geschrieben werden, kann der einzelne Bildspeicher Daten gleichzeitig mit der Leseoperation speichern. Jedoch werden die logischen Adressen der Schreibdaten nicht immer sequentiell von oben nach unten des logischen Raumes zugeführt. Angenommen ein Beispiel, wie in

Fig. 3B gezeigt, bei dem Bilddaten einer X-Achse ① zuerst in den Bildspeicher geschrieben werden, und dann Bilddaten einer Y-Achse ② geschrieben werden, und schließlich Bilddaten einer Linie ③ geschrieben werden. Dann können die Bilddaten der X-Achse erst in den Bildspeicher geschrieben werden, nachdem die Leseoperation von der oberen logischen Adresse (0, 0) des logischen Raumes zu der logischen Endadresse (X, Y) der X-Achse beendet ist. Deshalb können gleichzeitige Schreib- und Leseoperationen nicht immer ausgeführt werden. In diesem Fall muß die nächste Leseoperation oft erst nach dem Schreiben einer Seite von Bilddaten gestartet werden. Als Resultat kann eine ständige Leseoperation durch diesen einzelnen Bildspeicher nicht ausgeführt werden.

Figur 4 ist ein Blockdiagramm, das die Bilddatenverarbeitungsanordnung 3 des Datenverarbeitungssystems von Fig. 1 im allgemeinen ausführlich zeigt.

In Fig. 4 enthält die Bildspeichersteueranordnung die Bilddatenpufferschaltungsanordnung 30 (Fig. 1), einen Steuerprozessor (CTP) 31, einen Kanalcontroller (CHC) 32, einen Globalspeicher (GLM) 33, eine Bildhardware (IMH) 34, einen Druckcontroller (PRC) 35, einen Steuerbus (C-BUS) 36, einen Speicherbus (M-BUS) 37 und einen Bilddatenbus (I-BUS) 38. Unterbrochene Linien in Fig. 4 stellen Wege zum Übertragen von Steuersignalen dar. Durchgehende Linien stellen Wege zum Übertragen von Daten dar.

Der CTP 31 enthält einen lokalen Speicher (LM) 310, der aus einem Nur-Lese-Speicher (ROM) und einem Speicher mit wahlfreiem Zugriff (RAM) besteht, zum Speichern eines Steuerprogramms. Der RAM dient auch als Arbeitsbereich. Der CTP 31 steuert die gesamte Anordnung 3 und führt auch die Vorverarbeitung für die IMH 34 aus. Der CTP 31 kann durch einen Mikroprozessor realisiert sein.

Der CHC 32 ist durch den Hauptkanal 2 (Fig. 1) mit der

CPU 1 (Fig. 1) verbunden und führt eine Übertragungsoperation von Zeichendaten oder graphischen Daten von der CPU 1 zu dem GLM 33 durch ein Direktspeicherzugriffsverfahren (DMA) aus.

Der GLM 33 enthält einen Zwischenpuffer (IB) 330, einen Seitenpuffer (PB) 331, einen Graphikdatenpuffer (GDB) 332, einen Bilddatenpuffer (IDB) 333, einen Formularüberlagerungsspeicher (FOM) 334, einen Zeichenerzeugungsspeicher (CG) 335, einen Formularsteuerungspuffer (FCB) 336 und eine Zeichenumsetzungstabelle (TT) 337, von denen einige später ausführlicher beschrieben sind. Der GLM 33 hat eine Speicherkapazität in dem Bereich von zum Beispiel 0,5 Megabyte bis 2,5 Megabyte.

Die IMH 34 ist eine Hardware zum Konvertieren von Kodierten oder kompakten Daten, die in dem GLM 33 gespeichert sind, in Bitmusterdaten und zum Schreiben der Bitmusterdaten in den IMM 30, unter Steuerung des CTP 31. Die IMH 34 enthält einen Zeichengeneratorcontroller (CGC) 340, einen Vektorgeneratorcontroller (VGC) 341, einen Überlagerungsmustercontroller (OVC) 342 und einen Bildgeneratorcontroller (IMC) 343.

Der CGC 340 ist für die IMH 34 unentbehrlich und ist immer darin vorgesehen. Der CGC 340 hat die Funktion des Abbildens von Zeichenkodes, die in dem CG 335 gespeichert sind, in den GLM 33. Jeder der Zeichenkodes wird durch Konvertieren von Zeichendaten, die von der CPU 1 (Fig. 1) übertragen sind, in einen internen Ausdruck, d. h. einen Kode von vier Bytes pro Zeichen erhalten. Die Zeichenkodes werden von dem CG 335 in dem GLM 33 durch den M-BUS 37 zu dem CGC 340 durch DMA übertragen, wenn der CTP 31 den CGC 340 aktiviert. Die anderen Controller, d. h., VGC 341, OVC 342 und IMC 343, sind optional vorgesehen.

Der VGC 341 hat die Funktion des Abbildens von graphischen Datenkodes, die in dem GDB 332 gespeichert sind, in

den GLM 33. Die Daten, die dem VGC 341 zugeführt werden, sind graphische Codes, die durch Konvertieren graphischer Daten, die von der Host-CPU 1 (Fig. 1) übertragen wurden, in einen internen Ausdruck erhalten werden. Der VGC 341 greift auch auf den GDB 332 in dem GLM 33 durch den M-BUS 37 durch DMA zu, wenn der CTP 31 den VGC 341 aktiviert.

Der OVC 342 hat die Funktion des Wiederherstellens komprimierter Überlagerungsdaten, und der IMC 343 hat die Funktion des Wiederherstellens komprimierter Bilddaten.

Der C-BUS 36 ist ein Bus, der mit dem CTP 31 übereinstimmt. Als Beispiel besteht der C-BUS 36 aus einem 24-Bit-Adreßbus und einem 16-Bit-Datenbus. Alle Speicher und Register in der Bildspeichersteueranordnung 3 können innerhalb des Adreßraumes des C-BUS 36 aufgenommen werden.

Der C-BUS 36 kann 8-Bit-Daten oder 16-Bit-Daten übertragen.

Der M-BUS 37 ist ein Bus, der nur für den DMA von dem GLM 33 zu der IMH 34 verwendet wird, und kann unabhängig von dem C-BUS 36 arbeiten. Der M-BUS 37 besteht zum Beispiel aus einem 24-Bit-Adreßbus und einem 16-Bit-Datenbus. Der M-BUS 37 überträgt immer 16-Bit-Daten.

Der IMM 30 dient als Pufferspeicher zum Speichern von Bilddaten, die von der IMH 34 ausgegeben wurden. Die von der IMH 34 ausgegebenen Bilddaten sind Bitmusterdaten. Um das ständige Drucken zu erleichtern, ist der IMM 30 bei einer Ausführungsform der Erfindung in der Lage, zwei Seiten des logischen Speichers in nur einer Seite eines realen Speichers aufzunehmen. Die Bilddatenpufferschaltungsanordnung 30 kann zum Beispiel aus zwei individuellen Speicherpufferschaltungen oder "Ebenen" (IMM) 301 und 302 bestehen. Die erste Speicherebene 301 (nachstehend ausführlich beschrieben) kann für zwei Seiten von Zeichenbilddaten verwendet werden, die durch den CGC 340 entwickelt wurden. Die zweite Speicherebene 302 ist optional vorgesehen und kann für zwei Seiten von graphischen Bilddaten verwendet

werden, die durch den VGC 341 entwickelt wurden.

Der PRC 35 steuert eine Schnittstelle zwischen dem IMM 30 und einer Druckbaugruppe (in Fig. 4 nicht gezeigt). Synchron mit einer Zeitlage des Druckverfahrens in der Druckbaugruppe überträgt der PRC 35 die Bilddaten von dem IMM 30 zu der Druckbaugruppe. Wenn der IMM 30 aus zwei Speicherebenen besteht, sollte der PRC 35 ein entsprechend größeres Ausmaß haben.

Die Operation der Bilddatenverarbeitungsanordnung 3, die in Fig. 4 gezeigt ist, ist wie folgt. Der CHC 32 empfängt Zeichendaten von der Host-CPU 1 (Fig. 1) und überträgt sie dann durch den M-BUS 37 zu dem GLM 33 durch DMA. Der CTP 31 überwacht diese Datenübertragung sowie bildet den Inhalt des PB 331 in Übereinstimmung mit den empfangenen Zeichendaten. Sobald eine Seite des Inhalts in dem PB 331 gebildet ist, erzeugt der CTP 31 eine Anweisung für den CGC 340, so daß der CGC 340 die Zeichenkodes von dem PB 331 empfängt und ein Bitmuster der Zeichendaten entwickelt. Der IMM 301 speichert dann das entwickelte Bitmuster der Bilddaten. Wenn die Entwicklung des Bitmusters abgeschlossen ist, informiert der CGC 340 den CTP 31, daß die Entwicklung beendet ist. Dann erzeugt der CTP 31 eine Anweisung für den PRC 35, um Daten von dem IMM 301 zu lesen. Ansprechend auf die Anweisung liest der PRC 35 Daten von dem IMM 301 und gibt sie als Videosignale VS synchron mit der Operation der Druckbaugruppe aus.

Gleichzeitig mit der Lese- und Ausgabeoperation des PRC 35 zum Ausgeben der Videosignale VS wird die nächste Seite des Inhalts in dem PB 331 gebildet; der CGC 340 entwickelt ein Bitmuster von der nächsten Seite der Zeichendaten; und der IMM 301 speichert die nächste Seite des entwickelten Bitmusters auf ähnliche Weise wie bei der obengenannten Operation für die erste Seite.

In einem Fall, bei dem die zweite Speicherpufferschal-

tung IMM 302 vorhanden ist, wenn graphische Daten von der Host-CPU 1 übertragen werden, aktiviert der CTP 31 den VGC 341, nachdem eine Seite von graphischen Daten in dem PB 331 gespeichert ist, so daß der VGC 341 ein Bitmuster von einer Seite von graphischen Daten entwickelt. Der IMM 302 empfängt dann das Bitmuster der graphischen Daten. Die gleichzeitige Operation des Lesens der ersten Seite der graphischen Daten von dem IMM 302 und des Speicherns der nächsten Seite der graphischen Daten in den IMM 302 ist ähnlich der Operation für die Zeichendaten.

Wenn die Zeichendaten und die graphischen Daten auf derselben Seite ausgedruckt werden sollen, entnimmt der PRC 35 ein logisches ODER von den Ausgaben von den IMMs 301 und 302. Die Operation der Anordnung 3, die in Fig. 4 gezeigt ist, wird später unter Bezugnahme auf Fig. 15 bis 18 ausführlicher beschrieben.

Fig. 5 ist ein Blockschaltungsdiagramm, das die Bilddatenpufferschaltungsanordnung 301 gemäß einer Ausführungsform der vorliegenden Erfindung zeigt. In Fig. 5 ist der IMM 301 mit dem I-BUS 38 verbunden, der nur zum Zugreifen auf den IMM 301 oder IMM 302 (Fig. 4) verwendet wird. Der IMM 301 enthält eine Zellenadreuersetzungstabelle (CAT) 91, einen Zellenpeicher (CM) 92, eine Zellenadreuwarteschlange (CAQ) 93, eine erste Adreuersetzungsschaltung (EORa) 94, eine zweite Adreuersetzungsschaltung (EORb) 95, einen ersten Multiplexer (MPXa) 96, einen zweiten Multiplexer (MPXb) 97, einen dritten Multiplexer (MPXc) 98 und eine Bildspeichersteuerschaltung (IMC) 99.

Die CAT 91 ist ein logischer Speicher zum Speichern einer Tabelle, die verwendet wird, um eine logische Adresse in eine reale Adresse umzusetzen.

Der CM 92 ist ein realer Speicher zum Speichern von Bitbilddaten.

Die CAQ 93 ist eine "First-in-First-out"- (FIFO)

Schaltung zum Überwachen realer Speicherzellen, die nicht verwendet werden:

Die EORa 94 und die EORb 95 sind Schaltungen zum Umsetzen von Adressen und Datenfolgen, um einen seitlichen Zugriff, vertikalen Zugriff oder ebenen Zugriff zu realisieren.

$X_0$  bis  $X_{11}$  und  $Y_0$  bis  $Y_{11}$  stellen Adreßleitungen zum Spezifizieren einer logischen Lese- oder Schreibadresse des IMM 301 von der IMH 34 (Fig. 4) dar. P stellt eine Adreßleitung zum Spezifizieren einer Seite dar, die zum Lesen oder Schreiben verwendet wird.

$D_0$  bis  $D_{15}$  sind Datenleitungen zum Auslesen oder Einschreiben von Daten durch die IMH 34.

R/W ist ein Signal zum Unterscheiden, ob Daten durch die IMH 34 von dem IMM gelesen oder dahinein geschrieben werden.

RQ ist ein Zugriffsanforderungssignal von der IMH 34. ACK ist ein Zugriffserlaubnissignal von dem IMM 301 an die IMH 34.

Die anderen Signale in Fig. 5 sind später beschrieben.

Die Hauptoperation des in Fig. 5 gezeigten IMM 301 ist unter Bezugnahme auf Fig. 6 beschrieben, die gewisse Zustände der CAT 91, des CM 92, der CAQ 93 und eines logischen Raumes LS zeigt, der dem Zustand der CAT 91 entspricht. Der Einfachheit halber hat der logische Raum LS bei diesem Beispiel nur neun logische Zellen LC, die jeweils einen Adreßraum von 64 Bits x 64 Bits haben. Jede logische Zelle ist eine Einheit eines virtuellen Speichers. Auch der CM 92 hat neun reale Zellen RC, die jeweils denselben Adreßraum von 64 Bits x 64 Bits wie jenen der logischen Zelle LC haben. Die logischen Zellen LC und die realen Zellen RC können untereinander in einer Eins-zu-Eins-Korrespondenz in Beziehung stehen.

Die CAT 91 hat neun Spalten, die durch logische

Adressen YX = 00 bis 22 identifiziert sind. Jede Spalte enthält ein AOV-Bit, ein NOCELL-Bit und eine Adresse einer realen Zelle.

Das AOV-Bit gibt an, ob die logische Zelle einer  
5 logischen Zellenadresse zugeordnet werden kann oder nicht. Das heißt, im logischen Raum LS ist der schraffierte Teil, der die logischen Adressen YX = 02, 12, 20, 21 und 22 hat, als Beispiel des Zuordnungssperrbereichs, wie ein Bereich  
10 nahe den Rändern einer Seite gezeigt. Etwaige Daten in diesem Bereich sollten nicht ausgedruckt werden. Deshalb können die logischen Zellen LS in diesem Bereich keinen logischen Adressen zugeordnet werden. Zu diesem Zweck sind die AOV-Bits in den Spalten 02, 12, 20, 21 und 22 in der CAT 91 mit "1" gekennzeichnet, die anzeigt, daß die logischen  
15 Zellen LS in diesen Adressen für Schreib- oder Leseoperationen nicht verwendet werden können. Die anderen Spalten in der CAT 91 haben die AOV-Bits "0".

Das NOCELL-Bit zeigt an, ob zwischen der logischen Zelle und einer realen Zelle eine Korrespondenz besteht oder  
20 nicht. Bei dem dargestellten Beispiel entspricht die logische Zelle mit der logischen Adresse 00 der realen Zelle mit der logischen Adresse 00; die logische Zelle 01 entspricht der realen Zelle 01, die logische Zelle 10 entspricht der realen Zelle 02; und die logische Zelle 11  
25 entspricht der realen Zelle 04. So sind die NOCELL-Bits in den Spalten 00, 01, 10 und 11 mit "1" gekennzeichnet.

Bei diesem Beispiel ist die CAQ 93 ein FIFO mit neun Spalten. Jede Spalte speichert eine reale Adresse einer  
30 realen Zelle, die verwendet werden kann, um Daten hinein zu schreiben oder um Daten daraus zu lesen, d. h., die nicht in Gebrauch ist. Die Richtung des Datenstromes in der CAQ 93 geht von unten nach oben; wie in der Figur durch einen Pfeil angegeben. Leere Spalten in der CAQ 93 bedeuten, daß jede  
reale Adresse, außer den realen Adressen, die in der CAQ 93



gespeichert sind, mit einer logischen Adresse in Entsprechung gebracht werden kann. Bei dem dargestellten Zustand sind, da die realen Zellen 00, 01, 02 und 04 mit den logischen Zellen 00, 01, 02 bzw. 10 in Entsprechung gebracht worden sind, die übrigen realen Zellen 03, 05, 06, 07, 08 und 09 frei, um für andere Lese- oder Schreiboperationen verwendet zu werden. Deshalb können durch Vorsehen einer anderen CAT zusätzlich zu der CAT 91 Bilddaten in einem anderen logischen Raum in denselben CM 92 geschrieben werden.

Die Schreiboperation von zwei Seiten in eine einzelne Seite des realen Speichers ist unter Bezugnahme auf Fig. 7A bis 7C weiter beschrieben.

Figur 7A zeigt die Anfangszustände von zwei CATs, d. h., CAT ( $P_0$ ) für die erste Seite und CAT ( $P_1$ ) für die zweite Seite, und die CAQ. In diesem Zustand speichert die CAQ alle realen Zellenadressen 00 bis 08. Deshalb können alle realen Zellen für die Schreiboperation verwendet werden. Auch die NOCELL-Bits in der CAT ( $P_0$ ) und CAT ( $P_1$ ) sind alle "0", die anzeigt, daß keine logische Zelle irgendeiner der realen Zellen entspricht, auch wenn jede Spalte in der CAT ( $P_0$ ) und CAT ( $P_1$ ) irgendeine reale Adresse speichert.

Ausgehend von diesem, in Fig. 7A gezeigten Zustand wird angenommen, daß Bilddaten, wie in Fig. 7B gezeigt, auf die erste Seite  $P_0$  des logischen Raumes in einer Folge von ① bis ④ zu schreiben sind, d. h., in einer Folge der logischen Adressen 00, 01, 10 und 11 der ersten Seite  $P_0$ . Dann werden die realen Adressen 00, 01, 02 und 03, die in der CAQ gespeichert sind, sequentiell ausgegeben. Die ausgegebenen realen Adressen werden dann in die Spalten 00, 01, 10 und 11 der CAT ( $P_0$ ) eingegeben, wie in Fig. 7A durch Pfeile gezeigt. Als Resultat speichert die CAT ( $P_0$ ), wie in Fig. 7B gezeigt, die realen Adressen 00, 01, 02 und 03 in den Spalten 00, 01, 10 bzw. 11. Gleichzeitig mit der Speicher-

operation werden die NOCELL-Bits in diesen Spalten 00, 01, 10 und 11 zu "1" geändert. In diesem Zustand sind die Spalten 02, 12, 20, 21 und 22 der CAT ( $P_0$ ) und alle Spalten der CAT ( $P_1$ ) frei, um irgendeine reale Adresse zu speichern. Die CAQ speichert die realen Adressen 04, 05, 06, 07 und 08, die frei sind, um verwendet zu werden.

Ausgehend von dem in Fig. 7B gezeigten Zustand wird ferner angenommen, daß andere Bilddaten, wie in Fig. 7C gezeigt, in die zweite Seite  $P_0$  des logischen Raumes in einer Folge von ① bis ⑤ zu schreiben sind, d. h., in einer Folge der logischen Adressen 10, 11, 20, 22 und 02 der zweiten Seite  $P_1$ . Dann werden die realen Adressen 04 bis 08 von der CAQ sequentiell gelesen und in den Spalten 10, 11, 20, 21 bzw. 02 gespeichert. Gleichzeitig werden die NOCELL-Bits in diesen Spalten zu "1" geändert. Somit sind zwei Seiten von Bilddaten den realen Adressen einer einzelnen Seite des CM 92 zugeordnet.

Um die Bilddaten von dem CM 92 zu lesen, reicht es aus, die logische Adresse zu spezifizieren. Die Bilddaten in der entsprechenden realen Adresse, die der logischen Adresse zugeordnet ist, werden dann zu der Druckerbaugruppe gelesen.

Nach dem Lesen der Bilddaten wird die gelesene reale Zelle geöffnet, um verwendet zu werden, indem das NOCELL-Bit in der CAT 91 geändert wird. Deshalb ist während des Lesens einer Seite das gleichzeitige Schreiben in eine andere Seite möglich. Als Resultat wird eine ständige Leseoperation von der ersten Seite zu der zweiten Seite möglich.

Ferner kann, solange wenigstens eine reale Adresse in der CAQ gespeichert ist und wenigstens eine Spalte der CAT ( $P_1$ ) oder CAT ( $P_2$ ) das NOCELL-Bit von "0" hat, die reale Adresse in der Spalte gespeichert werden. Deshalb ist das Schreiben von Bilddaten in den logischen Raum nicht auf eine Operation von oben nach unten begrenzt. In Fig. 7C wird ein Kreis in einen unteren Teil des logischen Raumes geschrie-

ben, und dann wird ein Rechteck in einen oberen Teil des logischen Raumes geschrieben.

Figur 8 ist ein Flußdiagramm zum Erläutern der Operation der in Fig. 5 gezeigten Bilddatenpufferschaltungsanordnung. Unter Bezugnahme auf Fig. 5 und 8 wird bei Schritt 1201 die logische Adresse zum Zugreifen auf die CAT 91 durch die Daten  $X_6$  bis  $X_{11}$ ,  $Y_6$  bis  $Y_{11}$  und  $P$  bestimmt. Die niederwertigen sechs Bits  $X_0$  bis  $X_5$  und  $Y_0$  bis  $Y_5$  sind bei der Beschreibung unter Bezugnahme auf Fig. 8 nicht berücksichtigt, da eine logische Zelle aus 64 Bits x 64 Bits besteht. Die Daten  $P$  stellen die Seite  $P_0$  oder  $P_1$  dar. Dann wird durch die bestimmte logische Adresse auf die CAT 91 zugegriffen, so daß das AOV-Bit und das NOCELL-Bit, die der logischen Adresse entsprechen, gelesen werden.

Bei Schritt 1202 unterscheidet die IMC 99, ob das AOV-Bit "1" oder "0" ist.

Falls das AOV-Bit "1" ist, ist die bestimmte Adresse ein gesperrter Bereich für das Schreiben. Deshalb wird, falls die IMC 99 bei Schritt 1203 unterscheidet, daß das R/W-Signal von dem I-BUS 38 ein Schreibsignal ist, kein Verfahren ausgeführt. Falls das R/W-Signal ein Lesesignal ist, dann gibt die IMC 99 bei Schritt 1204 "0" aus, die durch die Multiplexer 98, 97, EORb 95, den Datenbus  $D_0$  bis  $D_{15}$  und den I-BUS 38 zu dem PRC 35 übertragen wird, so daß ein weißer Bereich gedruckt wird.

Falls das AOV-Bit bei Schritt 1202 "0" ist, unterscheidet die IMC 99 das NOCELL-Bit. Falls das NOCELL-Bit "0" ist, kann diese logische Adresse einer realen Adresse zugeordnet werden. Falls bei Schritt 1206 das R/W-Signal das Lesesignal ist, wird bei Schritt 1204 ein weißer Raum gedruckt, da die reale Zelle, die dieser logischen Adresse zugeordnet ist, gelesen worden ist und nicht in Gebrauch ist, da das NOCELL-Bit "0" ist. Falls jedoch das R/W-Signal das Schreibsignal ist, dann unterscheidet bei Schritt 1207 die IMC 99 durch

ein Signal EMP von der CAQ 93, ob die CAQ 93 leer ist oder nicht. Falls die CAQ 93 leer ist, kann keine reale Zelle verwendet werden, so daß bei Schritt 1208 die IMC 99 REJ auf "1" setzt. Das Zurückweisungssignal REJ wird durch den I-BUS 38 zu der IMH 34 übertragen. Als Antwort auf das Zurückweisungssignal REJ von "1" greift die IMH 34 wieder auf den IMM 301 zu.

Falls die CAQ 93 wenigstens eine reale Zelle bei Schritt 1207 speichert, dann erzeugt die IMC 99 bei Schritt 1209 ein Lesesignal QR, so daß die reale Adresse von der CAQ 93 gelesen wird.

Dann erzeugt die IMC 99 bei Schritt 1210 ein Schreibsignal CATWT, so daß die entnommene reale Adresse in der logischen Adresse der CAT 91 gespeichert wird. Gleichzeitig wird das NOCELL-Bit in dieser logischen Adresse auf "1" geändert.

Dann erzeugt bei Schritt 1211 die IMC 99 ein Schreibsignal CMWT, so daß, durch die spezifizierte logische Adresse, Daten von der IMH 34 durch den I-BUS und Datenbus D<sub>0</sub> bis D<sub>15</sub> in die reale Zelle geschrieben werden, die der spezifizierten logischen Adresse zugeordnet ist.

Falls das NOCELL-Bit bei Schritt 1205 "1" ist, bedeutet dies, daß eine reale Zelle der spezifizierten logischen Adresse zugeordnet worden ist. Deshalb wird bei Schritt 1212 die reale Adresse, die der spezifizierten logischen Adresse zugeordnet ist, von der CAT 91 gelesen.

Dann wird bei Schritt 1213, falls das R/W-Signal das Schreibsignal ist, Schritt 1211 ausgeführt. Falls das R/W-Signal das Lesesignal ist, werden dann bei Schritt 1214 Daten von der realen Adresse des CM 92 gelesen. Die gelesenen Daten werden durch den Datenbus D<sub>0</sub> bis D<sub>15</sub> und den I-BUS 38 zu dem PRC 35 übertragen.

Bei Schritt 1215 unterscheidet die IMC 99, ob ein DUMP-Signal von dem PRC 35 übertragen ist oder nicht. Das DUMP-

Signal wird erzeugt, nachdem jede reale Zelle gelesen ist.  
Das DUMP-Signal bedeutet beim seitlichen Zugreifen  
(nachstehend unter Bezugnahme auf Figuren 12 und 13  
erläutert), daß die Daten in einer realen Zelle, die 64  
5 horizontale Linien ( $Y_0 - Y_5$ ) hat, vollständig gelesen sind.  
Das DUMP-Signal bedeutet auch beim vertikalen Zugreifen  
(nachstehend unter besonderer Bezugnahme auf Figur 18B  
erläutert), daß die Daten in einer realen Zelle, die 64  
vertikale Linien ( $X_0 - X_5$ ) hat, vollständig gelesen sind.  
10 Falls das DUMP-Signal empfangen ist, erzeugt die IMC 99 ein  
Schreibsignal CAWT, so daß die reale Adresse der realen  
Zelle, die bei dem Schritt 1214 gelesen wurde, in die CAQ 93  
geschrieben wird. Dann wird bei Schritt 1217 das NOCELL-Bit  
in der spezifizierten logischen Adresse der CAT 91 zu "0"  
15 geändert.

Figur 9 ist ein Logikschaltungsdiagramm der IMC 99 in  
dem IMM 301, der in Fig. 5 gezeigt ist. In Fig. 9 enthält  
die IMC 99 ODER-Gatter 131 und 136 und UND-Gatter 132 bis  
135 und 137 und 138. Die IMC 99 empfängt die Signale NOCELL,  
20 AOV, R/W, DUMP und EMP und gibt die Signale BLNK, QR, QWT,  
CATWT, CMWT und REJ aus. Das ODER-Gatter 131 und das UND-  
Gatter 132 führen das Verfahren der Schritte 1203 bis 1208  
in dem in Fig. 8 gezeigten Flußdiagramm aus. Die UND-Gatter  
133 bis 136 führen die Schritte 1209 bis 1211 und die  
25 Schritte 1215 bis 1217 aus. Die UND-Gatter 137 und 138  
führen die Schritte 1202, 1203, 1206 und 1207 aus.

Figur 10 zeigt die Struktur der CAT 91 in dem in Fig. 5  
gezeigten IMM 301. In Fig. 5 besteht jede logische Adresse  
LOG.ADD der CAT 91 aus  $P$ ,  $Y_{11}$  bis  $Y_6$  und  $X_{11}$  bis  $X_6$ . Die  
30 Gesamtanzahl von Bits in jeder logischen Adresse beträgt  
somit 13. Jede logische Adresse LOG.ADD entspricht einer  
logischen Zelle LC in den logischen Räumen LS. Jede logische  
Zelle LC besteht aus 64 Bits x 64 Bits. Eine Seite des  
logischen Raumes LS besteht aus  $64 \times 64 = 4096$  logischen

Zellen LC. Deshalb bestehen zwei Seiten des logischen Raumes aus etwa 8 K logischen Zellen. Demzufolge beträgt die Anzahl von logischen Adressen der CAT 91 8 K Wörter, wovon jedes aus 13 Bits besteht. Somit umfaßt die CAT 91 effektiv zwei  
5 Adreßkorrespondenzspeicher zur jeweiligen Verwendung beim Lesen/Schreiben von bilddefinierenden Daten von zwei verschiedenen Bildern.

Figur 11 zeigt die Struktur des CM 92 in dem in Fig. 5 gezeigten IMM 301. In Fig. 11 besteht jede reale Adresse des  
10 CM 92 aus 19 Bits, d. h.,  $A_3$  bis  $A_0$  von der EORa 94,  $A_7$  bis  $A_4$  von dem I-BUS 38 und  $A_{18}$  bis  $A_8$  von der CAT 91 durch den MPXa 96. Bei der vorhergehenden Beschreibung unter Bezugnahme auf Fig. 8 sind die niederwertigen Bits der logischen Adresse oder realen Adresse nicht berücksichtigt. In der  
15 Praxis jedoch werden die niederwertigen Bits verwendet, um aus den Bilddaten ein Bitmuster in dem CM 92 zu entwickeln und, insbesondere, um den seitlichen Zugriff oder vertikalen Zugriff zu realisieren, wie später ausführlicher beschrieben.

20 Der CM 92 besteht aus 16 Speicherchips  $CP_0$  bis  $CP_{15}$ . Jeder Speicherchip besteht aus 512 K Bits. Deshalb besteht der CM 92 aus  $512 \text{ K} \times 16 = 8000 \text{ K}$  Bits, d. h., 512 K Wörtern, wobei ein Wort aus 16 Bits besteht. Da eine reale Zelle RC aus 256 Wörtern besteht, besteht der CM 92 bei  
25 diesem Beispiel aus 2 K realen Zellen. Es sei daran erinnert, daß der logische Raum für eine Seite aus 4 K Zellen besteht, wie unter Bezugnahme auf Fig. 10 beschrieben. Der Grund, daß die Anzahl von realen Zellen kleiner als die Anzahl von logischen Zellen ist, liegt darin, daß die  
30 Größe der Druckseite nicht immer denselben Adressenraum wie den logischen Raum erfordert. Wenn zum Beispiel das Druckpapier unterschiedliche Längs- und Querseitenlängen hat, erfordert der reale Adressenraum nicht das gesamte Quadrat der längeren Seite.

Die höherwertigen Adreßsignale  $A_{18}$  bis  $A_8$  werden durch Adressenkonvertierung aus den Adreßsignalen  $X_6$  bis  $X_{11}$ ,  $Y_6$  bis  $Y_{11}$  und  $P$  durch die CAT 91 und die CAQ 93 erhalten, wie zuvor unter Bezugnahme auf Fig. 6 bis 8 beschrieben. Die  
5 mittelwertigen Adreßsignale  $A_7$  bis  $A_4$  werden direkt aus den Adreßsignalen  $X_4$ ,  $X_5$ ,  $Y_4$  und  $Y_5$  erhalten. Diese Adressen sind für alle Speicherchips  $CP_0$  bis  $CP_{15}$  gemeinsam spezifiziert.

Die niederwertigen Adreßsignale  $A_3$  bis  $A_0$  sind jedoch  
10 für entsprechende Speicherchips  $CP_0$  bis  $CP_{15}$  unterschiedlich. Diese Adreßsignale  $A_3$  bis  $A_0$  werden durch Konvertieren der Adressen  $X_3$  bis  $X_0$  und  $Y_3$  bis  $Y_0$  durch die EORa 94 erhalten, wie später ausführlich beschrieben. Diese Konvertierung ist notwendig, um den seitlichen Zugriff, vertikalen  
15 Zugriff oder ebenen Zugriff auszuführen.

Ein spezifisches Beispiel einer vertikalen Zugriffsoperation, die durch die Schaltungsanordnung von Fig. 4  
ausgeführt wird, wird später unter Bezugnahme auf Fig. 18B beschrieben, aber mit kurzen Worten, falls der vertikale  
20 Zugriff ausgeführt wird, ist eine Zugriffsoperation auf einen Bereich von 1 Bit in X-Richtung x 16 Bits in Y-Richtung in dem CM 92 auszuüben. Wie jedoch aus der Struktur des in Fig. 11 gezeigten CM 92 ersichtlich ist, sind die 16  
Bits in Y-Richtung alle in demselben Speicherchip enthalten,  
25 und es ist unmöglich, auf mehrere Adressen auf demselben Speicherchip gleichzeitig zuzugreifen.

Durch Ausführen der obengenannten Adressenkonvertierung wird das Zugreifen auf ein Wort in den seitlichen, vertikalen oder ebenen Zugriffsmodi möglich.

30 Die seitlichen, vertikalen und ebenen Zugriffsoperationen als solche sind in der japanischen Patentanmeldung Nr. 58-117405 erörtert. Hier wird eine kurze Erläuterung des seitlichen Zugriffs und vertikalen Zugriffs unter Bezugnahme auf Fig. 12 und 13 gegeben.

Figur 12 ist ein Blockdiagramm, das die EORa 94 und die EORb 95 in dem in Fig. 5 gezeigten IMM 301 ausführlich zeigt. In Fig. 12 enthält die EORa 94 einen Multiplexer 161 und eine Adressenanordnungsschaltung (AR) 162. Die EORb 95 enthält eine Adressenumordnungsschaltung (RAR) 163 und sechzehn Multiplexer 164-1 bis 164-16.

Der Multiplexer 161 empfängt die Adreßsignale  $X_0$  bis  $X_3$ ,  $Y_0$  bis  $Y_3$  und ein Funktionssignal FUNK, das anzeigt, ob der Zugriff auf den CM ein vertikaler Zugriff (1 Bit x 16 Bits), horizontaler Zugriff (16 Bits x 1 Bit) oder ebener Zugriff (4 Bits x 4 Bits) ist. Ansprechend auf das Funktionssignal FUNK wählt der Multiplexer ein Ausgangssignal  $R_0$  bis  $R_3$  aus. Die Beziehung zwischen dem Funktionssignal FUNK und den Ausgangssignalen  $R_0$  bis  $R_3$  ist in einer Tabelle in Fig. 12 gezeigt. Das heißt, wenn das Funktionssignal FUNK 00 ist, das anzeigt, daß ein vertikaler Zugriff ausgeführt werden sollte, dann wählt der Multiplexer 161 die Adreßsignale  $X_0$  bis  $X_3$  als die Ausgangssignale  $R_0$  bis  $R_3$  aus. Wenn FUNK 01 ist, das anzeigt, daß ein ebener Zugriff ausgeführt werden sollte, wählt der Multiplexer 161 die Adreßsignale  $X_0$ ,  $X_1$ ,  $Y_0$  und  $Y_1$  als die Ausgangssignale  $R_0$  bis  $R_3$  aus. Wenn FUNK 11 ist, das anzeigt, daß ein horizontaler Zugriff ausgeführt werden sollte, wählt der Multiplexer 161  $Y_0$  bis  $Y_3$  als  $R_0$  bis  $R_3$  aus.  $R_0$  bis  $R_3$  werden der AR 162 und der RAR 163 zugeführt. Die AR 162 erzeugt gemäß einer vorbestimmten Regel und ansprechend auf FUNK sechzehn Adressen als niederwertige Adressen ( $A_0$  bis  $A_3$ ) x 16 zum Zugreifen auf entsprechende Speicherchips  $CP_0$  bis  $CP_{15}$ . Die sechzehn Adressen ( $A_0$  bis  $A_3$ ) x 16 unterscheiden sich voneinander, wie unter Bezugnahme auf Fig. 11 erwähnt. Somit sind die niederwertigen Adressen ( $A_0$  bis  $A_3$ ) x 16 für den vertikalen, seitlichen oder ebenen Zugriff bestimmt.

Um Daten in den CM 92 zu schreiben oder um Daten daraus zu lesen, erzeugt die RAR gemäß einer anderen vorbestimmten



Regel und ansprechend auf FUNK sechzehn Datenauswahlsignale. Jedes der Datenauswahlsignale besteht aus vier Bits. Diese Datenauswahlsignale werden jeweilig den Multiplexern 164-1 bis 164-16 zugeführt. So wählen die Multiplexer 164-1 bis 164-16 jeweils entsprechende Daten aus, die von dem MPXb 97 (Fig. 9) übertragen wurden, als Daten, die auszugeben sind. Die von den Multiplexern 164-1 bis 164-16 ausgegebenen Daten werden dann zu dem CM 92 übertragen, um dahinein geschrieben zu werden, oder zu dem I-BUS 38 als Lesedaten.

Durch die oben beschriebene Anordnung und Umordnung der Adreßsignale  $R_0$  bis  $R_3$  konstruieren die Adressen von 16 Bits für eine Zugriffsoption immer ein Wort aus sechzehn Speicherchips  $CP_0$  bis  $CP_{15}$ .

Figur 13A ist ein Diagramm zum Erläutern einer Zuführungsrichtung eines Druckpapiers  $PA_1$ , zum Drucken eines sogenannten Hochformats. Figur 13B ist ein Diagramm zum Erläutern einer anderen Zuführungsrichtung eines Druckpapiers  $PA_2$ , zum Drucken eines sogenannten Querformats.

Die Druckpapiere  $PA_1$  und  $PA_2$  haben oft unterschiedliche Längs- und Querseitenlängen. Die Zuführungsrichtung in beiden Fällen ist auf Grund der Trommelgröße der Druckerbaugruppe gewöhnlich senkrecht zu der kürzeren Seite des Druckpapiers.

Für das Hochformat werden die Zeichen und Bilder in die Richtung parallel zu der kürzeren Seite des Druckpapiers  $PA_1$  gedruckt, wie in Fig. 14A gezeigt. Für das Querformat werden die Zeichen und Bilder in die Richtung parallel zu der längeren Seite des Druckpapiers  $PA_2$  gedruckt, wie in Fig. 14B gezeigt.

Beim Schreiben von Daten im Hochformat ist der horizontale Zugriff zweckmäßig, und beim Schreiben von Daten im Querformat ist der vertikale Zugriff zweckmäßig.

In Fig. 13A und 13B ist der logische Raum LC als Quadrat gezeigt. Der Punkt 0 auf der oberen linken Seite des

logischen Raumes LS ist der Ursprung des logischen Raumes LS. Da die Seitenlänge der Druckseite  $PA_1$  oder  $PA_2$  kürzer als die Seite des quadratischen logischen Raumes ist, ist der schraffierte Bereich nicht nötig. Deshalb sind die  
5 logischen Adressen in dem schraffierten Bereich mit AOV = "1" versehen, wie vorher beschrieben. Durch dieses einfache Verfahren wird keine reale Adresse diesen logischen Adressen zugeordnet, die mit AOV = "1" versehen sind. Als Resultat kann der reale Speicher ein kleinerer Adreßraum als der  
10 logische Raum gewesen sein, wie unter Bezugnahme auf Fig. 11 vorher erwähnt.

Sowohl bei Hochformat als auch bei Querformat liest der PRC 35 (Fig. 4) den CM 92 (Fig. 5) in die Richtung senkrecht zu der Papierzuführungsrichtung.

15 Es erfolgt nun eine ausführliche Beschreibung der Bilddatenverarbeitungsanordnung 3, die in Fig. 4 gezeigt ist, unter Bezugnahme auf Figs. 15 bis 18.

Figur 15A ist ein Flußdiagramm zum Erläutern des Datenstromes von dem CHC 32 zu der IMH 34 in der in Fig. 4  
20 gezeigten Anordnung 3. Unter Bezugnahme auf Fig. 4 und 15A empfängt der CTP 31 bei Schritt 501 einen Anweisungskode IC und Daten D von der CPU 1 durch den CHC 32. Der Anweisungskode IC hat ein Format, das als ein Beispiel in Fig. 15B gezeigt ist. Der Anweisungskode bei diesem Beispiel besteht  
25 aus einem Byte, d. h., 8 Bits von 0 bis 7. Der Hauptinhalt des Anweisungskodes zeigt an, ob die Daten von der CPU 1 Zeichendaten CD oder graphische Daten GD sind, oder stellt eine Anweisung dar, um auf den Inhalt des FCB 336 (Fig. 4) zuzugreifen, der die vertikale Position und den Zeilen-  
30 abstand auf einer Seite bestimmt, wenn die Daten von der CPU 1 Zeichendaten sind. In Fig. 15B, wenn nur das höchste signifikante Bit 7 von den höherwertigen vier Bits "1" ist, zeigt der Anweisungskode an, daß die Daten von der CPU 1 Zeichendaten sind. In diesem Fall stellen die niederwertigen

vier Bits 0 bis 3 eine Kanalzahl CH.NO dar, die eine Zeile auf einer Seite bestimmt, d. h., die vertikale Position des Zeichens, das auf einer Seite auszudrucken ist. Wenn nur das nächste signifikante Bit 6 "1" ist, zeigt der Anweisungskode an, daß die Daten von der CPU 1 graphische Daten sind. Wenn nur das Bit 5 "1" ist, stellt der Anweisungskode eine Anweisung dar, um auf den FCB 334 zuzugreifen. Ein Anweisungskode mit einem anderen Bitmuster kann eine Anweisung ausgeben, um die Druckerbaugruppe zu betreiben.

10 Bei Schritt 502 speichert der CTP 31 die Daten von der CPU 1 in den IB 330 in dem GLM 33 ohne Abwandlung der empfangenen Daten.

Bei Schritt 503 analysiert der CTP 31 den Anweisungskode IC, um zu bestimmen, ob die Daten Zeichendaten CD oder graphische Daten GD sind.

15 Falls die empfangenen Daten graphische Daten GD sind, dann wird bei Schritt 504 die Verarbeitung der graphischen Daten durch einen Graphikgenerator ausgeführt, der den GDB 332, den VGC 341 und den IMM 302 enthält.

20 Falls die empfangenen Daten Zeichendaten CD sind, dann wird bei Schritt 505 die vertikale Position des Zeichens, das auf einer Zeile einer Seite auszudrucken ist, unter Bezugnahme auf den FCB 334 auf der Grundlage des empfangenen Anweisungskodes bestimmt.

25 Das Format des FCB 334 ist als Beispiel in Fig. 15C gezeigt. Der Inhalt des FCB 334 ist gemäß dem gewünschten Druckformat vorher festgelegt. Angenommen, daß die Länge einer Seite 50 12 Zoll beträgt, und daß der minimale Zeilenabstand zwischen benachbarten Zeilen 12 Zeilen/Zoll beträgt.

30 Dann beträgt die maximale Zeilenanzahl auf einer Seite 144 Zeilen, 0 bis 143. Demzufolge besteht der FCB 334, der die vertikale Position jeder Zeile auf der Seite festlegt, maximal aus 144 Spalten, "0" bis "143", wobei jede Spalte einer Zeile auf der Seite 50 entspricht. Jede Spalte in dem

FCB 334 besteht aus Zeilenabstandssteuerdaten LP, die den Zeilenabstand und eine Kanalzahl CH.NO, die eine zu überspringende Zeile anzeigt, angeben. Die Kanalzahl CH.NO ist eine von "0" bis "12". Gemäß dem gewünschten Druckformat können in verschiedenen Spalten dieselben Kanalzahlen vorgesehen sein. Der CTP 31 zählt die Zahl aufwärts, 5  
wieviele Male dieselben Kanalzahlen empfangen sind. Wenn zum Beispiel der Anweisungskode zuerst die Kanalzahl "1" in dem in Fig. 15C gezeigten FCB 334 spezifiziert, sucht der CTP 10  
31, beginnend bei der Spalte "0", die erste "1" der Kanalzahl. Da die erste Kanalzahl "1" in der Spalte "3" bei diesem Beispiel gespeichert ist, bewegt der CTP 31 seinen Zeiger, um die Zeile "3" zu steuern. Angenommen, daß dann der Anweisungskode wieder die Kanalzahl "1" spezifiziert, 15  
sucht der CTP 31 die zweite "1" der Kanalzahl. Die zweite Kanalzahl "1" ist in der Spalte "5" gespeichert. Deshalb springt der Zeiger des CTP 31 von der Spalte "3" zu der Spalte "5". Falls der nächste Anweisungskode die Kanalzahl "7" spezifiziert, bewegt der CTP 31 seinen Zeiger von der 20  
Spalte "5" zu der Spalte "143", falls die Spalten "0" bis "142" die Kanalzahl "7" nicht speichern.

Der Zeilenabstand LP ist einer von "0", "1" und "2", die jeweilig darstellen, daß der Abstand zwischen benachbarten Zeilen 6 Zeilen/Zoll, 8 Zeilen/Zoll und 12 Zeilen/Zoll beträgt. Angenommen, daß die Auflösung des Druckers 240 25  
Punkte/Zoll beträgt. Dann werden die Zeichendaten auf der ersten Zeile "0", falls durch den Anweisungskode spezifiziert, auf der Position entwickelt, die um 40 Punkte vom oberen Seitenrand vertikal niedriger liegt, da die erste 30  
Spalte "0" den Zeilenabstand "0" enthält, d. h., 6 Zeilen/Zoll, und deshalb benötigt eine Zeile 40 Punkte in vertikale Richtung. Auch die Zeichendaten auf der zweiten Zeile "1" werden, falls durch den Anweisungskode spezifiziert, auf der Position entwickelt, die um 30 Punkte von der

ersten Zeile "0" vertikal niedriger liegt, da die zweite Spalte "1" den Zeilenabstand "1" enthält, d. h., 8 Zeilen/Zoll, und deshalb benötigt eine Zeile 30 Punkte in vertikale Richtung.

5        Unter erneuter Bezugnahme auf Fig. 4 und 15A liest der CTP 31 nach dem Schritt 505 die Zeichendaten Byte für Byte von dem IB 330 in dem GLM 33. Die gelesenen Daten werden als hexadezimaler erweiterter binär verschlüsselter Dezimalaustauschkode (EBCDIC) eines Zeichenkodes verarbeitet. Das  
10        heißt, bei Schritt 506 wird der Lesezeichenkode in eine obere Adresse des Zeichenmusters umgesetzt, das in dem CG 335 des GLM 33 gespeichert ist, unter Bezugnahme auf die Zeichenumsetzungstabelle (TT) 337 in dem GLM 33.

      Die Beziehung zwischen dem CG 335 und der TT 337 ist in  
15        Fig. 15D gezeigt. In Fig. 15D besteht die TT aus 256 Spalten. Jede Spalte speichert die obere Adresse eines Zeichenmusters, das dem Zeichenkode (EBCDIC) entspricht. Der CG 335 speichert Zeichenmuster für Zeichen A, B, C, ... . Wenn als Beispiel ein Zeichenkode C.C. von "01" in dem  
20        hexadezimalen Ausdruck von dem IB 330 gelesen ist, sucht der CTP 31 die Spalte "1" der TT 337, mit dem Resultat, daß die obere Adresse  $TA = a_{t1}$  des Zeichens "A" identifiziert ist. Die identifizierte obere Adresse  $a_{t1}$  wird bei Schritt 507 in dem PB 331 gespeichert. Ähnlich wird, wenn ein Zeichenkode  
25        von "03" spezifiziert ist, die obere Adresse  $TA = a_{t2}$  des Zeichens B identifiziert und bei Schritt 507 in dem PB 331 gespeichert.

      Figur 15E zeigt das Format des Seitenpuffers (PB) 331. In Fig. 15E werden anders als die obengenannten oberen  
30        Adressen TA der Zeichenmuster Bewegungsbeträge MA in horizontaler Richtung und Adressen PDMA zum Zugreifen auf einen Druckdatenspeicher (siehe Fig. 16) in dem CGC 371 bei Schritt 507 auch in dem PB 331 gespeichert. Der Druckdatenspeicher wird später ausführlicher beschrieben.

Die Bewegungsbeträge MA sind unter Bezugnahme auf Fig. 15F erläutert. Wie in Fig. 15F gezeigt, sind die Bewegungsbeträge die Abstände von den Druckstartpositionen von Zeichen auf einer Seite in horizontaler Richtung. Die Druckstartposition für jede Zeile auf einer Seite ist in dem Druckdatenspeicher gespeichert. In Fig. 15F ist zum Beispiel der Bewegungsbetrag für das Zeichen "A" "0"; der Bewegungsbetrag für das Zeichen "B" ist " $S_1$ "; und der Bewegungsbetrag für das Zeichen "C" ist " $S_2$ ". Wenn die Größe eines Zeichens in horizontaler Richtung als 10 Zeichen/Zoll definiert ist, und wenn die Auflösung 240 Punkte/Zoll beträgt, dann beträgt der Abstand in horizontaler Richtung zwischen einer Startadresse eines Zeichens und einer Startadresse des nächsten Zeichens 24 Punkte. In diesem Fall ist  $S_1$  24 und  $S_2$  ist 48.

Nachdem durch Wiederholen der Schritte 506 bis 508 auf obengenannte Weise alle Zeichenkodes für eine Seite verarbeitet sind, setzt der CTP 31 bei Schritt 509 eine Endemarke EF in den PB 331. In diesem Zustand speichert der PB 331 die Zeicheninformationen für alle Zeichen, die auf einer Seite auszudrucken sind, die die Bewegungsbeträge MA, die oberen Adressen TA der Zeichenmuster und die Adressen PMDA zum Zugreifen auf den Druckdatenspeicher enthalten.

Dann setzt der CTP 31 bei Schritt 510 die nötigen Parameter in den CGC 340. Diese Parameter sind zum Entwickeln von Bitmusterdaten aus den Zeichendaten erforderlich.

Schließlich fordert der CTP 31 bei Schritt 511 den CGC auf, aus den Zeichendaten Bitmusterdaten zu entwickeln.

Der CGC 340 beginnt dann zu arbeiten.

Figur 16 zeigt ein Blockdiagramm des CGC 340 in der in Fig. 4 gezeigten Anordnung. In Fig. 16 enthält der CGC 340 den Druckdatenspeicher (PDM) 61, wie zuvor erwähnt, ein Steuerregister 62, einen CGC-Controller (CONT) 63, eine CG-

Zugriffsschaltung (ACS) 64 und eine Schreibschaltung (WC) 65.

Der PDM 61 und das Steuerregister 62 speichern nötige Parameter, wie zuvor unter Bezugnahme auf Fig. 15A und 15E und, insbesondere, bei Schritt 510 in Fig. 15A beschrieben. Präziser speichert der PDM 61 Startadressen von entsprechenden Zeilen, die auf eine Seite zu drucken sind, und eine Vielzahl von Zeichenparametern, die eine Marke enthalten, die anzeigt, ob das zu druckende Zeichen gedreht werden sollte oder nicht, und eine Breite und eine Höhe für jedes Zeichen anzeigt. Das Steuerregister 62 enthält ein Befehlsregister (C.REG) 621 zum Speichern eines Startbefehls zum Starten der Operation des CGC 340, ein Statusregister (S.REG) 622 zum Speichern von Statusdaten des CGC 340, ein Speicheradreßregister (MA.REG) 623 zum Anzeigen einer oberen Adresse TA in dem PB 331 und ein Zeilenadreßregister (L.REG) 624 zum Speichern einer vertikalen Position (siehe FCB in Fig. 15C) des zu druckenden Zeichens. Diese Daten in dem PDM 61 und in dem Steuerregister 62 werden unter der Steuerung des CTP 31 in der in Fig. 4 gezeigten Anordnung 3 gesetzt.

Nachdem der CTP 31 den Inhalt des PB 331 setzt, die Daten in dem PDM 61 bereitstellt, die obere Adresse des PB 331 in dem M.A. REG 623 setzt und die vertikale Position in dem L.REG 624 setzt, schreibt dann der CTP 31 einen Startbefehl in das Befehlsregister 621. Ansprechend auf den Startbefehl von dem Befehlsregister 621 beginnt CONT 63 wie folgt zu arbeiten:

(1) Unter Verwendung der oberen Adresse des PB 331, die in dem Speicheradreßregister 623 gespeichert ist, liest der CONT 63 die in dem PB 331 gespeicherten Daten. Der PB 331 hat vorher die Daten, die die oberen Adressen TA der Zeichenmuster in dem CG 335, die Bewegungsbeträge MA und die Adressen des PDM 61 enthalten, für eine Seite gespeichert, wie zuvor unter Bezugnahme auf Fig. 15D, 15E und 15F.

beschrieben. Die von dem PB 331 gelesenen Daten enthalten dieselbe obere Adresse TA wie die obere Adresse, die von dem Speicheradreßregister 623 übertragen wurde.

- (2) Die obere Adresse TA, die von dem PB 331  
5 gelesen wurde, wird durch den CONT 63 zu der ACS 64  
übertragen.

Der CONT 63 berechnet die Position des zu  
schreibenden Zeichens auf der Grundlage der Daten der  
Startadresse einer Zeile, die in dem PDM 61 gespeichert ist,  
10 und des Bewegungsbetrages MA, die in den von dem PB 331  
gelesenen Daten enthalten sind.

Unter Verwendung der Adresse PDMA des PDM 61,  
wobei die Adresse in den von dem PB 331 gelesenen Daten  
enthalten ist, liest der CONT 63 ferner die entsprechenden  
15 Zeichenparameter von dem PDM 61 und überträgt die gelesenen  
Zeichenparameter zu der WC 65.

Dann aktiviert der CONT 63 die WC 65, um  
ihren Betrieb zu starten.

- (3) Dann sendet die WC 65 ein CG-Anforderungs-  
20 signal zu der ACS 64. Die ACS 64 greift dann auf den CG 335  
zu, um eine Linie des Zeichenmusters "A", zum Beispiel, zu  
lesen, das durch die Daten von dem PB 331 spezifiziert ist.  
Die gelesene eine Linie des Zeichenmusters wird in den IMM  
301 geschrieben. Bei der Schreiboperation wird ein vertika-  
25 les Schreiben oder ein horizontales Schreiben ausgeführt,  
wie später ausführlicher beschrieben. Nachdem eine Linie des  
Zeichenmusters von dem CG 335 gelesen ist, inkrementiert die  
ACS 64 die Adresse zum Zugreifen auf den CG 335.

- (4) Die Operation wie im obigen Punkt (3)  
30 beschrieben wird für jede Linie des Zeichenmusters wieder-  
holt. Wenn das Zeichenmuster vollständig in den IMM 301  
geschrieben ist, informiert die WC 65 den CONT 63, daß die  
Schreiboperation des Zeichenmusters beendet ist.

- (5) Dann greift der CONT 63 wieder auf den PB 331



zu, um nachfolgende Daten für das nächste zu schreibende Zeichen zu lesen, und die Operation ähnlich jener, die oben in den Punkten (1) bis (4) beschrieben ist, wird wiederholt.

Die obige Operation wird sequentiell wiederholt, bis  
5 die Endemarke EF von dem PB 331 zu dem CONT 63 gelesen ist. Wenn die Endemarke EF gelesen ist, informiert der CONT 63 den CTP 31 (Fig. 4) durch das Statusregister 622, daß die Schreiboperation für eine Seite beendet ist.

Figur 17 zeigt ausführlich die Schreibschaltung (WC)  
10 65, die in dem in Fig. 16 gezeigten CGC enthalten ist. In Fig. 17 enthält die WC 65 ein Parallel-Seriell-Konvertierungsschieberegister (SRa) 650, ein Seriell-Parallel-Konvertierungsschieberegister (SRb) 651, einen X-Zähler 652, einen Y-Zähler 653, einen H-Zähler 654, einen V-Zähler 655, eine  
15 Schreibsteuerschaltung (WCC) 656 und einen Multiplexer 657. Der X-Zähler 652 ist ein Zähler zum Überwachen der X-Adressen des IMM 301; der Y-Zähler 653 ist ein Zähler zum Überwachen der Y-Adressen des IMM 301; der H-Zähler 654 ist ein Zähler zum Überwachen der Breite eines Zeichens; und der  
20 V-Zähler 655 ist ein Zähler zum Überwachen der Höhe eines Zeichens.

Figur 18A ist ein Flußdiagramm zum Erläutern der Operation der in Fig. 17 gezeigten WCC 656. In Fig. 18A und in der folgenden Beschreibung der Operation der WCC 656 ist  
25 hauptsächlich ein horizontales Zugriffsverfahren beschrieben. In dem Flußdiagramm sind alle vertikalen Zugriffsverfahren in Klammern gesetzt. Die Operation wird in der folgenden Reihenfolge ausgeführt:

(1) Zu Beginn, nachdem die WCC 656 von dem CONT 63  
30 (Fig. 16) eine Schreibposition  $X_1$  und  $Y_1$  ( $X_2$  und  $Y_2$  bei einem vertikalen Zugriff), eine Breite H und eine Höhe V eines zu druckenden Zeichens empfangen hat, und nachdem der CONT 63 die WCC 656 aktiviert hat, um ihre Operation zu starten, dann setzt die WCC 656 bei Schritt 801 diese Werte

$X_1$ ,  $Y_1$ , H und V in die X-, Y-, H- bzw. V-Zähler 652 bis 655.

(2) Wenn das SRa 650 durch die WCC 656 als leer erkannt wird, dann sendet sie bei Schritt 802 ein Anforderungssignal REQ an die ACS 64 (Fig. 16), das Zeichendaten in dem CG 335 anfordert.

(3) Bei Schritt 803 verzögert die WCC 656 ihre Operation, bis 16 Bits der gelesenen Zeichendaten in dem SRa 650 gesetzt sind. Wenn die gelesenen Daten in dem SRa 650 gesetzt sind, wird die Adresse der ACS 64 (Fig. 16) zum Zugreifen auf den CG 335 inkrementiert.

(4) Bei Schritt 804 löscht die WCC 656 das SRb 651.

(5) Bei Schritt 805 wird ein Bit der Daten, die von dem SRa 650 seriell ausgegeben wurden, in das SRb 651 unter der Steuerung der WCC 656 verschoben.

(6) Bei Schritt 806 inkrementiert die WCC 656 den X-Zähler 652 (dekrementiert den Y-Zähler 653) und dekrementiert den H-Zähler 654.

(7) Bei Schritt 807 unterscheidet die WCC 656, ob das SRb 651 voller Daten ist oder nicht, d. h., ob die niederwertigen vier Bits des X-Zählers 652 bei dem horizontalen Zugriffsverfahren alle "1" sind oder nicht. Das heißt, die WCC 656 unterscheidet, ob 16 Bits von Daten, d. h., eine Dateneinheit, die eine Zugriffseinheit zum Zugreifen auf den IMM 301 ist, in das SRb 651 verschoben sind oder nicht. Auch bei Schritt 807 unterscheidet die WCC 656, ob der H-Zähler 654 leer ist oder nicht, d. h., ob die Menge von Daten, die von dem CG 335 übertragen wurden, die volle Breite H des Zeichens erreicht.

Falls beide Antworten "NEIN" sind, werden dann die Schritte 805 bis 807 wiederholt.

Falls wenigstens eine der Antworten "JA" ist, dann wird Schritt 808 ausgeführt.

(8) Bei Schritt 808 unterscheidet die WCC 656, ob der

gesamte Inhalt in dem SRb 651 "0" ist oder nicht. Falls der gesamte Inhalt "0" ist, wird kein Zugreifen zum Schreiben ausgeführt, und das Verfahren geht über zu Schritt 814. Da keine Schreiboperation ausgeführt wird, wenn der gesamte Inhalt des SRb 651 "0" ist, wird eine Hochgeschwindigkeitsverarbeitung realisiert, wenn das Zeichenmuster einen großen weißen Bereich enthält. Falls das SRb 651 wenigstens ein Bit von "1" enthält, dann wird Schritt 809 ausgeführt.

5 (9) Bei Schritt 809 erzeugt die WCC 656 ein Anforderungssignal REQ zum Speicherschreibzugriff. Das Anforderungssignal REQ wird dem IMM 301 zugeführt.

10 (10) Bei Schritt 810 wartet die WCC 656, bis ein Zugriffserlaubnissignal ACK von dem IMM 301 übertragen ist. Dieses Warten ist nur nötig, falls eine andere Anordnung den I-BUS 38 (Fig. 4) verwendet.

15 (11) Bei Schritt 811 werden die Daten  $D_0$  bis  $D_{15}$  in dem SRb 651, die Daten  $X_0$  bis  $X_{11}$  in dem X-Zähler 652 und die Daten  $Y_0$  bis  $Y_{11}$  in dem Y-Zähler 653 durch den I-BUS 38 zu dem IMM 301 übertragen. So werden 16 x 1 Bit von Daten in den IMM 301 auf eine logische Schreibadresse geschrieben, die auf der Grundlage der Daten  $X_0$  bis  $X_{11}$  und  $Y_0$  bis  $Y_{11}$  bestimmt ist, wie zuvor ausführlicher beschrieben. Falls jedoch keine der logischen Adressen für die Schreiboperation geöffnet sind, wie zuvor ausführlich beschrieben wurde, dann kann die logische Schreibadresse nicht bestimmt werden, bis wenigstens eine logische Adresse zum Schreiben geöffnet ist. In diesem Fall erzeugt der IMM 301 ein Zurückweisungssignal.

20 (12) Bei Schritt 812 unterscheidet die WCC 656, ob das Zurückweisungssignal von dem IMM 301 übertragen ist oder nicht. Falls es übertragen ist, geht das Verfahren zu Schritt 813 über; falls nicht, geht das Verfahren zu Schritt 814 über.

25 (13) Bei Schritt 813 hält das Verfahren für eine angemessene Zeitperiode an, um die Belegung des I-BUS 38 zu

vermeiden. Nach jener Zeitperiode werden die Schritte 810 bis 812 wiederholt.

(14) Bei Schritt 814 unterscheidet die WCC 656, ob der H-Zähler leer ist oder nicht. Falls eine Linie des Zeichens nicht verarbeitet ist, geht das Verfahren dann zu Schritt 804 über.

(15) Falls eine Linie des Zeichens verarbeitet worden ist, geht das Verfahren zu Schritt 815 über; bei dem unterscheidet die WCC 656, ob der V-Zähler 655 leer ist oder nicht. Falls er leer ist, ist das Zeichen vollständig in den IMM 301 geschrieben worden. Falls er nicht leer ist, geht das Verfahren zu Schritt 816 über.

(16) Bei Schritt 816 werden der X-Zähler (Y-Zähler) und der H-Zähler für die nächste Linie in dem Zeichen zurückgesetzt; der Y-Zähler (X-Zähler) wird inkrementiert; und der V-Zähler wird dekrementiert. Dann geht das Verfahren zu Schritt 802 über.

Der Grund dafür, daß die zwei Schieberegister SRa 650 und SRb 651 in der WC 65 vorgesehen sind, wird unter Bezugnahme auf Fig. 18B verständlich, in der Schreibpositionen auf dem logischen Raum des IMM 301 sowohl für den seitlichen Zugriff als auch für den vertikalen Zugriff dargestellt sind. Die Breite H des Zeichens A ist als 32 Bits dargestellt. Das SRa 650 empfängt zuerst 16 Bits von dem CG 335, wie durch schraffierte Linien angegeben. Die obere Adresse des Zeichenmusters A ist  $(X_1, Y_1)$  für den seitlichen Zugriff oder  $(X_2, Y_2)$  für den vertikalen Zugriff. Wie in Fig. 18B gezeigt, stimmt die X-Adresse  $X_1$  oder die Y-Adresse  $Y_2$  nicht immer mit dem höchsten signifikanten Bit MSB der 16 Bits überein. Bei dem Beispiel ist die X-Adresse  $X_1$  oder die Y-Adresse  $Y_2$  von dem MSB um 6 Bits entfernt. Da die WCC 656 die X-Adresse  $X_1$  oder die Y-Adresse  $Y_2$  kennt, speichert das SRb 651 die höherwertigen 6 Bits nicht, und nur die übrigen 10 Bits werden in das SRb 651 sequentiell verschoben, von

dem höherwertigen Bit, d. h., Bit 9, zu dem niederwertigen Bit, d. h., Bit 0. So gibt das SRb 651 nur 10 Bits bei der ersten Schreiboperation aus. Da die höherwertigen 6 Bits von den empfangenen 16 Bits vernachlässigt werden, erhöht sich  
5 die Betriebsgeschwindigkeit. Bei der zweiten Schreiboperation werden alle 16 Bits von dem SRa 650 zu dem SRb 651 verschoben. Bei der dritten Schreiboperation werden jedoch nur die höherwertigen 6 Bits von den empfangenen 16 Bits in das SRb 651 verschoben.

10       Figur 19 ist ein Blockdiagramm, das den allgemeinen Aufbau des in Fig. 4 gezeigten PRC 35 zeigt. In Fig. 19 enthält der PRC 35 eine IMM-Schnittstellensteuerschaltung (INT CONT) 191, eine Adreßsteuerschaltung 192, eine Modussteuerschaltung (MOD CONT) 193, ein Videosteuerungs-UND-Gatter 194, ein Schieberegister (SR) 195 und ein Datenpuffer-  
15 register (DBR) 196.

Der PRC 35 sendet verschiedene Steuersignale von der INT CONT 191, der ADD CONT 192 und der MOD CONT 193 durch den I-BUS 38 zu dem IMM 30 (Fig. 4). Das DBR 196 empfängt  
20 Bilddaten (IMD) von dem IMM 30 durch den I-BUS 38. Die einmal in dem DBR 196 gespeicherten Daten werden zu dem SR 195 übertragen. Das SR 195 konvertiert dann die Bilddaten in serielle Daten, in Übereinstimmung mit einem Steuersignal von der INT CONT 191. Die seriellen Daten von dem SR 195  
25 werden durch das UND-Gatter 194 gegattert. So wird das Videosignal VS als logisches Produkt der seriellen Daten und eines Videosteuerungssignals (VCS) von der INT CONT 191 erhalten.

Die INT CONT 191 sendet das Speicherauszugssignal DUM, wenn der PRC eine reale Zelle gelesen hat, um die reale  
30 Zelle zu öffnen, d. h., damit sie zum Einschreiben von neuen Daten verwendet werden kann.

Die MOD CONT 193 steuert die Leserichtung, d. h., Hochformat oder Querformat, durch Erzeugen eines Modus-

signals MOD. Die ADD CONT 192 steuert die Leseadresse.

Figur 20 zeigt den Aufbau der INT CONT 191. In Fig. 20 zählt ein Linienadrezähler (CNT) 201 eine Scannlinie (64 Zähler). Auf der Grundlage des Zählwertes erzeugt ein Signalgenerator (SIG GEN) 202 Zeitlagendaten IBACC zum Übertragen von Adreßsignalen und Modusdaten, wobei die Zeitlagendaten der ADD CONT 192 und der MOD CONT 193 (Fig. 19) zugeführt werden, das Datenanforderungssignal REQ, das Videosteuerungssignal VCS und das Datensetzzeitlagensignal DST dem DBR 196 und dem SR 195 (Fig. 19) zugeführt werden.

Eine Speicherauszug-Anweisungsschaltung (DUM) 203 erzeugt auch das Speicherauszugssignal DUM auf der Grundlage der Adressen  $X_0$  bis  $X_5$  und  $Y_0$  bis  $Y_5$  zum Zugreifen auf den IMM 30.

Figur 21A zeigt den Aufbau der Adreßsteuerschaltung (ADD CONT) 192, die in Fig. 19 gezeigt ist. In Fig. 21A werden die Leseadressen  $X_0$  bis  $X_{11}$  und  $Y_0$  bis  $Y_{11}$  wie folgt erhalten.

Wenn das Modussignal MOD das Hochformat spezifiziert, wie in Fig. 13A gezeigt, wird ein Zählwert "0" anfangs von einem Multiplexer 211 in einen Aufwärts-/Abwärtszähler 212 geladen, zum Erzeugen der X-Adressen  $X_0$  bis  $X_{11}$ . Dann steuert ein Aufwärts-/Abwärtssteuersignal U/D CONT den U/D CNT 211, um ansprechend auf ein Taktsignal CLK X, das eine Datensetzzeitlage (CLK 1) darstellt, um sechzehn aufwärts zu zählen. Währenddessen zählt ein Zähler (CNT) 213 ansprechend auf ein Scannen einer Linie (64 Zähler) von null um eins aufwärts (CLK Y = CLK 2).

Wenn das Modussignal MOD das Querformat spezifiziert, wie in Fig. 13B gezeigt, wird ein Wert, der die Breite in X-Richtung des Druckpapiers darstellt, von einem Register (REG) 214 in den U/D CNT 212 geladen, zum Erzeugen der X-Adressen. Der Wert wird von dem C-BUS 36 übertragen und ist vorher in dem Register 214 gespeichert. Dann zählt der U/D

CNT 212 ansprechend auf das Scannen einer Linie (64 Zähler) um eins abwärts ( $\text{CLK X} = \text{CLK 2}$ ). Der CNT 213 für die Y-Adressen zählt ansprechend auf die Datensetzzeitlage um sechzehn aufwärts ( $\text{CLK Y} = \text{CLK 1}$ ).

5       Figur 21B zeigt eine Logikschaltung zum Erzeugen der Taktsignale CLK X und CLK Y. Wie aus Fig. 21B ersichtlich ist, ist bei dem Hochformatausdruck MOD "1" und \*MOD ist "0", so daß CLK X gleich CLK 1 ist und CLK Y gleich CLK 2 ist. Bei dem Querformatausdruck ist \*MOD "1" und MOD ist  
10       "0", so daß CLK X gleich CLK 2 ist und CLK Y gleich CLK 1 ist.

Die vorliegende Erfindung ist nicht auf die vorhergehende Beschreibung der Ausführungsform begrenzt, und verschiedene Veränderungen und Abwandlungen sind innerhalb  
15       des Umfanges der Erfindung möglich.

Obwohl zum Beispiel die Erläuterung des IMM 30 hauptsächlich bezüglich des Zeichengeneratorcontrollers (CGC) 340 erfolgt, kann der Grundgedanke der Zuordnung der realen Zellen zu logischen Adressen bei einer Ausführungsform der  
20       Erfindung ähnlich angewendet werden, wenn der Vektorgeneratorcontroller (VGC) 341, der Überlagerungsmustercontroller (OVC) 342 oder der Bildgeneratorcontroller (IMC) 343 aktiviert ist.

Ferner ist die Einheit der realen Zelle oder logischen  
25       Zelle nicht auf das beschriebene Beispiel von 64 Bits x 64 Bits begrenzt, sondern kann jede Maßeinteilung haben, die 1 Bit x 1 Bit enthält.

Ferner ist noch in der vorhergehenden Beschreibung die CAQ 93 als eine "First-in-First-out-" (FIFO) Schaltung  
30       beschrieben. Jedoch kann anstelle der CAQ 93 jeder Speicher zum Speichern der realen Adressen verwendet werden, die nicht benutzt werden, solange ihre Eingabefolge und Ausgabe-  
folge unter einer gewissen Bestätigungsbedingung gesteuert werden.

5 So kann eine Bilddatenpufferschaltungsanordnung vorgesehen sein, bei der eine einzelne Bildspeicherebene zum gleichzeitigen Lesen und Schreiben von verschiedenen Seiten verwendet werden kann, auch wenn das Schreiben nicht der Reihenfolge nach ausgeführt wird.

10 Aus der vorhergehenden Beschreibung geht hervor, daß eine Ausführungsform der Erfindung eine verbesserte Datenpufferschaltungsanordnung vorsehen kann, bei der die Speicherkapazität des realen Speichers verringert werden  
15 kann, ohne die Lesegeschwindigkeit zu reduzieren, und bei der das Schreiben von Bilddaten in eine Seite eines logischen Raumes nicht auf eine sequentielle Operation (von oben nach unten) begrenzt ist, so daß neue Bilddaten auf jede Position auf der Seite geschrieben werden können,  
20 solange wenigstens eine reale Zelle gerade nicht benutzt wird, um vorhandene Daten zu speichern. Bei einer bevorzugten Ausführungsform der Erfindung kann das ständige Lesen von verschiedenen Seiten in dem logischen Raum unter Verwendung eines einzelnen realen Speicherraumes erreicht werden, d. h., ohne Verwendung von verschiedenen realen Speicherräumen.

Dementsprechend können Ausführungsformen dieser Erfindung besonders vorteilhaft auf dem Gebiet der Hochgeschwindigkeitsübertragung von Bilddaten von einem  
25 Bildspeicher zum Beispiel zu einem Laserdrucker sein.



ANSPRÜCHE

1. Eine Bilddatenpufferschaltungsanordnung mit:
  - einem Speicher (CM 92), der eine Vielzahl von individuell adressierbaren Datenspeicherplätzen (RC) hat, zum Speichern von Daten, die ein Bild definieren, das in eine Vielzahl von räumlichen Einheiten (LC) unterteilt ist;
  - einem Adreßspeichermittel (CAQ 93) zum Speichern von Adressen (REAL ADD) von solchen Plätzen (RC), die zum Speichern von neuen Daten verfügbar sind;
  - einem Adreßkorrespondenzspeicher (CAT 91), der Speicherbereiche hat, die jeweilig den genannten räumlichen Einheiten (LC) entsprechen und mittels entsprechender Adreßkodes (LOG ADD) zugänglich sind;
  - einem Schreibmittel zum Schreiben der bilddefinierenden Daten, falls vorhanden, von irgendeiner ausgewählten der genannten räumlichen Einheiten (LC) in einen individuellen von solchen verfügbaren Plätzen (RC), welcher verfügbarer Platz durch Entnahme der Adresse (REAL ADD) jenes Platzes (RC) aus dem Adreßspeichermittel (CAQ 93) ausgewählt worden ist, und zum Ablegen der Adresse (REAL ADD) jenes ausgewählten Platzes (RC) in dem individuellen Speicherbereich des Korrespondenzspeichers (CAT 91), der der gewählten räumlichen Einheit (LC) entspricht; und
  - einem Lesemittel zum Verwenden des Adreßkodes (LOG ADD), der für den Speicherbereich passend ist, der einer ausgewählten der genannten räumlichen Einheiten (LC) entspricht, um aus dem genannten Speicher (CM 92) die bilddefinierenden Daten, falls vorhanden, der ausgewählten räumlichen Einheit (LC) auszulesen, und zum Wiederhinzufügen der Adresse (REAL ADD) des Speicherplatzes (RC), aus dem jene bilddefinierenden Daten ausgelesen worden sind, zu dem genannten Adreßspeichermittel (CAQ 93).

2. Eine Schaltungsanordnung nach Anspruch 1, die zwei solche Adreßkorrespondenzspeicher (CAT 91) zur jeweiligen Verwendung beim Lesen/Schreiben von bilddefinierenden Daten von zwei verschiedenen Bildern hat, wobei die Operation der Schreib- und Lesemittel so ist, daß Daten, die eines der Bilder definieren, in den genannten Speicher (CM 92) geschrieben werden können, während Daten, die das andere Bild definieren, aus jenem Speicher gelesen werden.
3. Eine Schaltungsanordnung nach Anspruch 1, bei der die Anzahl der genannten Speicherbereiche in dem genannten Adreßkorrespondenzspeicher (CAT 91) wenigstens das Zweifache der Anzahl der genannten Datenspeicherplätze (RC) in dem genannten Speicher (CM 92) beträgt.
4. Eine Schaltungsanordnung nach Anspruch 1, 2 oder 3, bei der jedem der genannten Speicherbereiche in dem Korrespondenzspeicher (CAT 91) ein individuelles korrespondenzanzeigendes Bit (NOCELL) zugeordnet ist, zum Anzeigen, ob bilddefinierende Daten der entsprechenden räumlichen Einheit (LC) in dem Speicher (CM 92) an einem Platz gespeichert sind oder nicht, dessen Adresse in dem betreffenden Speicherbereich gehalten ist, wobei die Schaltungsanordnung so ist, daß, wenn die bilddefinierenden Daten einer ausgewählten räumlichen Einheit (LC) in den Speicher (CM 92) geschrieben sind, das genannte korrespondenzanzeigende Bit (NOCELL), das dem entsprechenden Speicherbereich zugeordnet ist, durch das Schreibmittel gesetzt wird, und daß, wenn die bilddefinierenden Daten jener räumlichen Einheit (LC) aus dem Speicher ausgelesen sind, das genannte korrespondenzanzeigende Bit (NOCELL), das dem entsprechenden Speicherbereich zugeordnet ist, durch das Lesemittel zurückgesetzt wird.

5. Eine Schaltungsanordnung nach Anspruch 4, bei der das genannte Lesemittel beim Feststellen, daß das genannte korrespondenzanzeigende Bit, das einem Speicherbereich zugeordnet ist, der zum Lesen ausgewählt ist, im Rücksetzzustand ist, ein Leerdatensignal als bilddefinierende Daten für die entsprechende räumliche Einheit (LC) erzeugt.

6. Eine Schaltungsanordnung nach irgendeinem vorhergehenden Anspruch, bei der das genannte Adreßspeichermittel (CAQ 93) eine First-in-First-out-Schaltung ist.

7. Eine Schaltungsanordnung nach irgendeinem vorhergehenden Anspruch, bei der der genannte Korrespondenzspeicher (CAT 91) auch individuelle korrespondenzsperrende Bits (AOV) hat, die jeweilig den genannten Datenspeicherbereichen zugeordnet sind, welche Bits gesetzt werden können, um zu verhindern, daß die bilddefinierenden Daten der entsprechenden räumlichen Einheiten in den genannten Speicher geschrieben werden.

8. Eine Schaltungsanordnung nach Anspruch 7, bei der das genannte Lesemittel beim Feststellen, daß das Korrespondenzsperrende Bit (AOV), das einem Speicherbereich zugeordnet ist, der zum Lesen ausgewählt ist, gesetzt ist, ein Leerdatensignal als bilddefinierende Daten für die entsprechende räumliche Einheit (LC) erzeugt.

Fig. 1

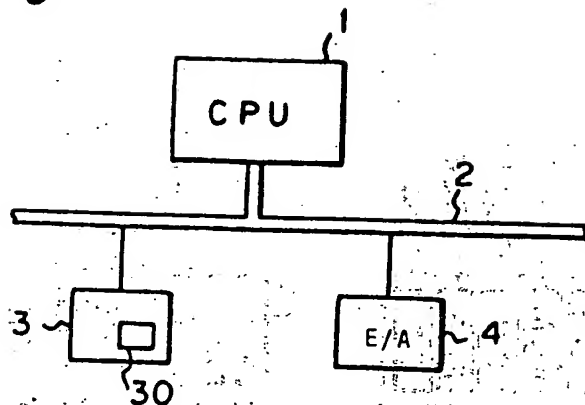


Fig. 2

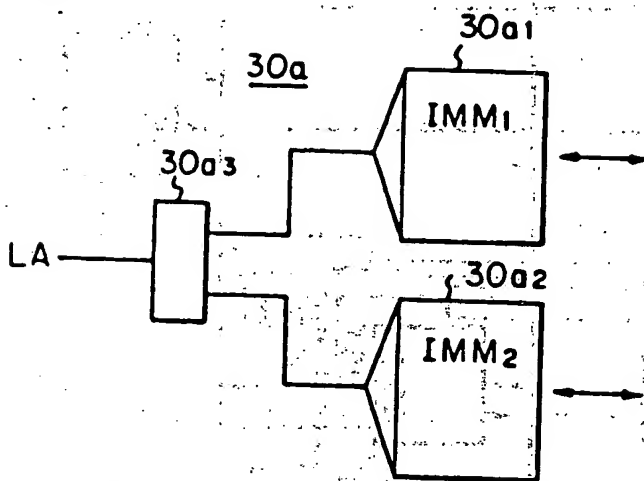


Fig. 3A

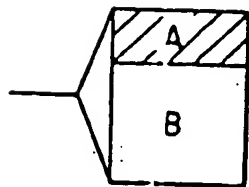
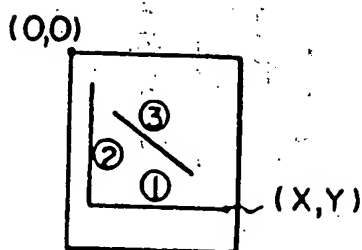


Fig. 3B



2/22

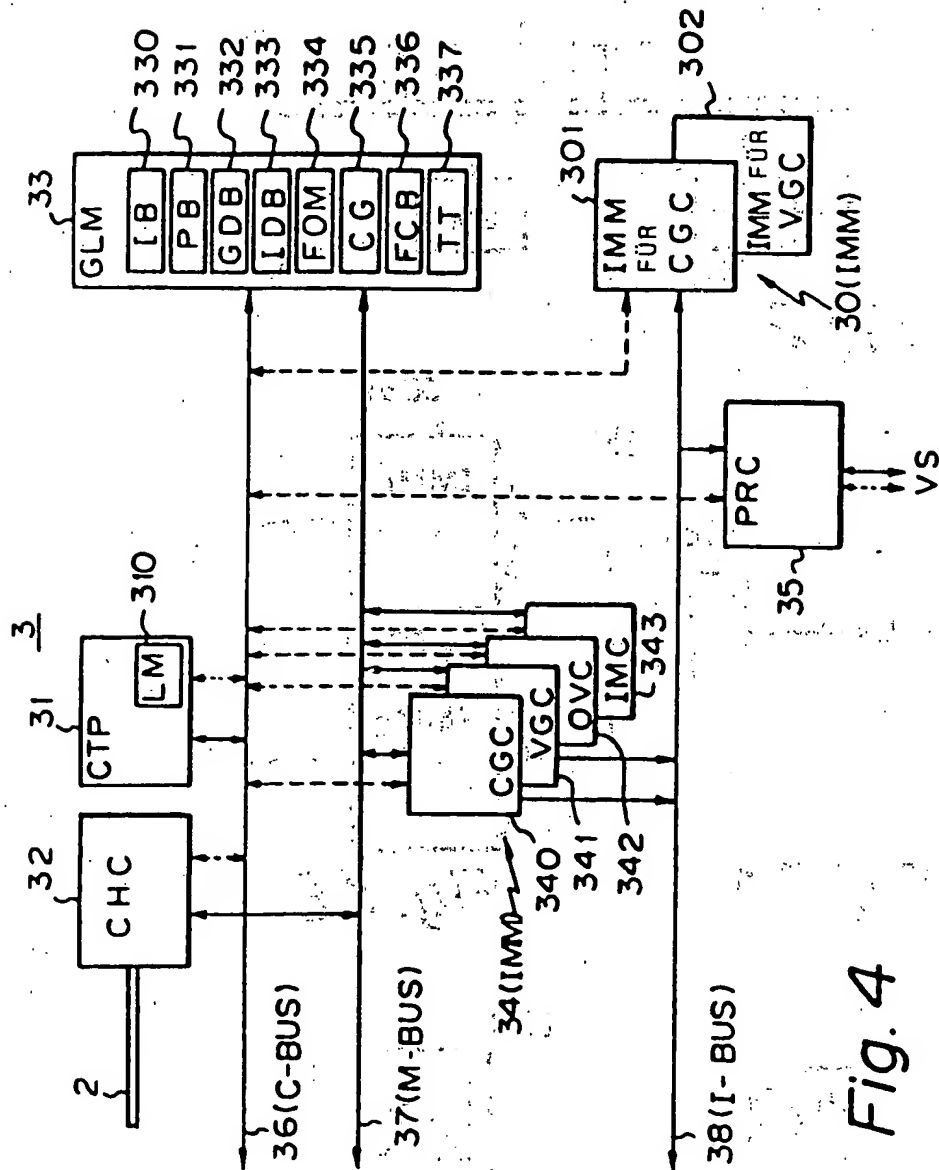
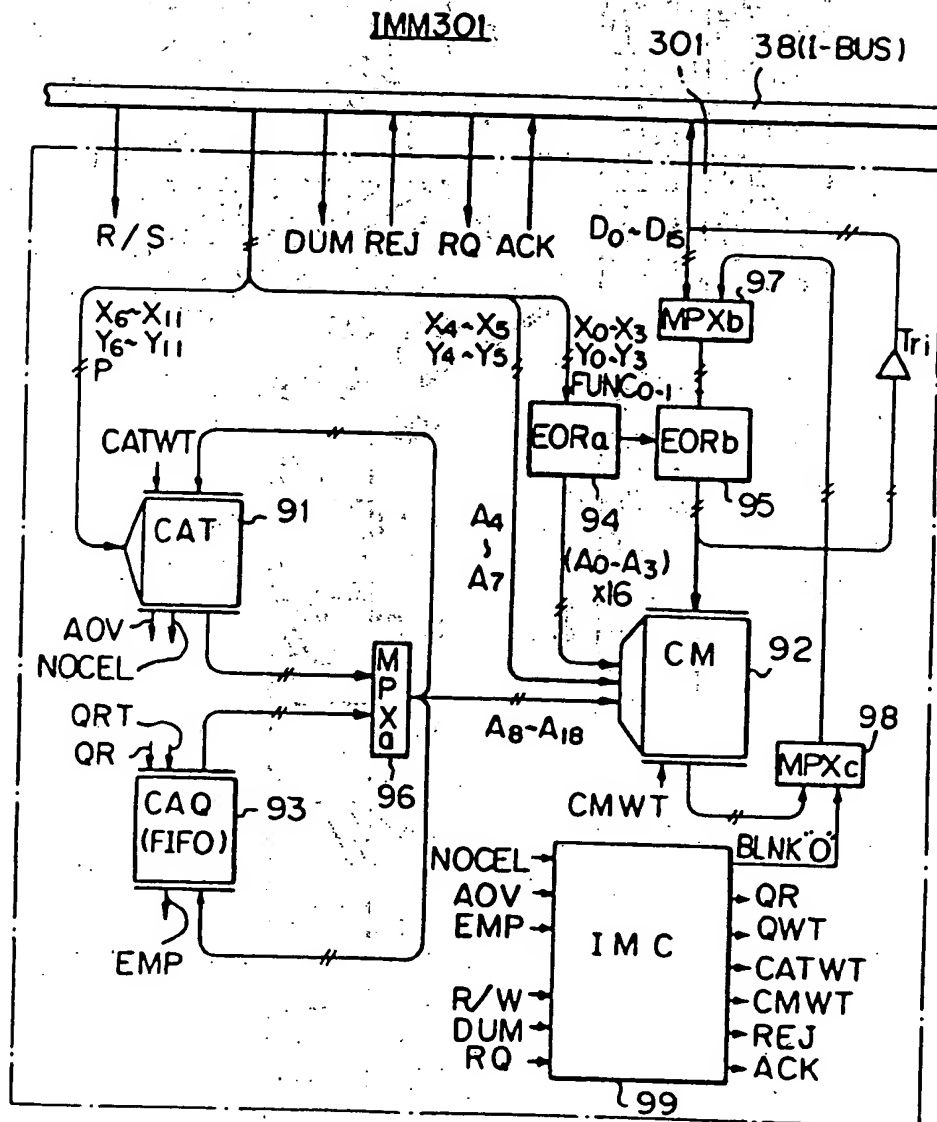


Fig. 4

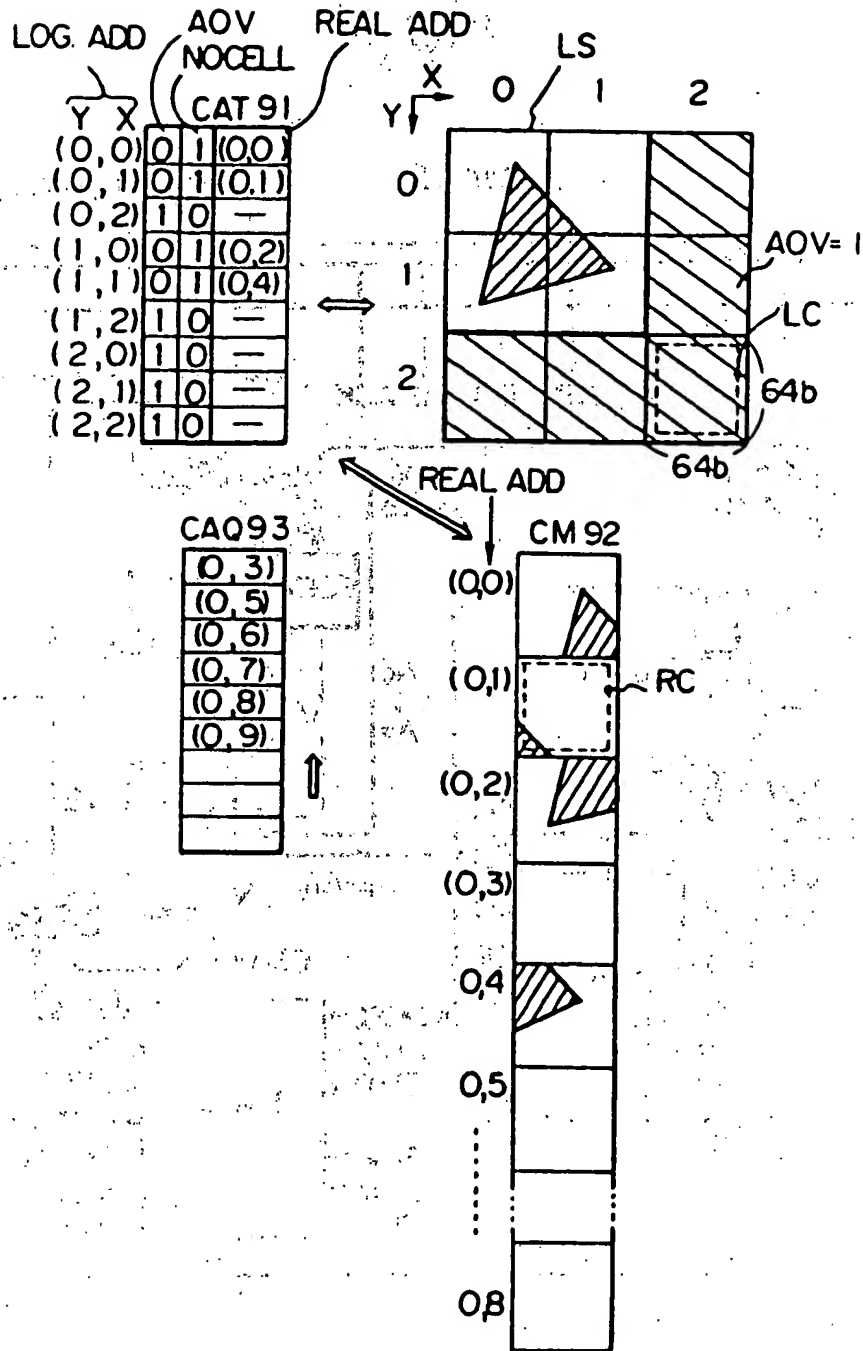
3/22

Fig. 5



4/22

Fig. 6



5/22

Fig. 7A

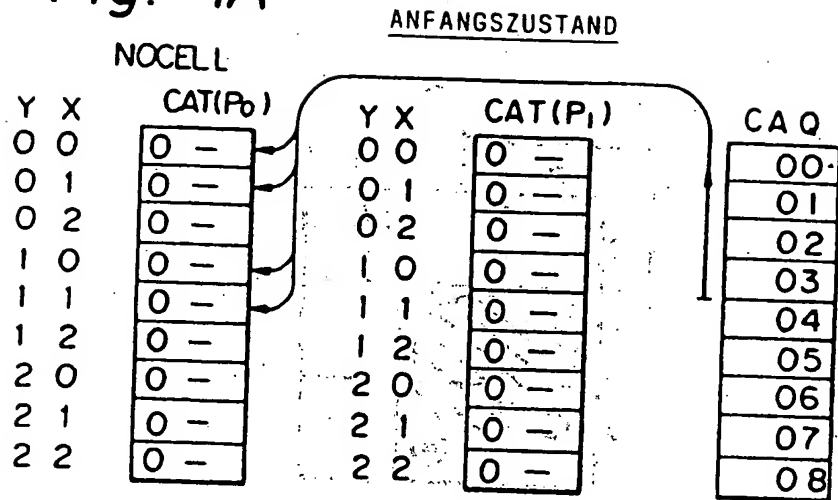
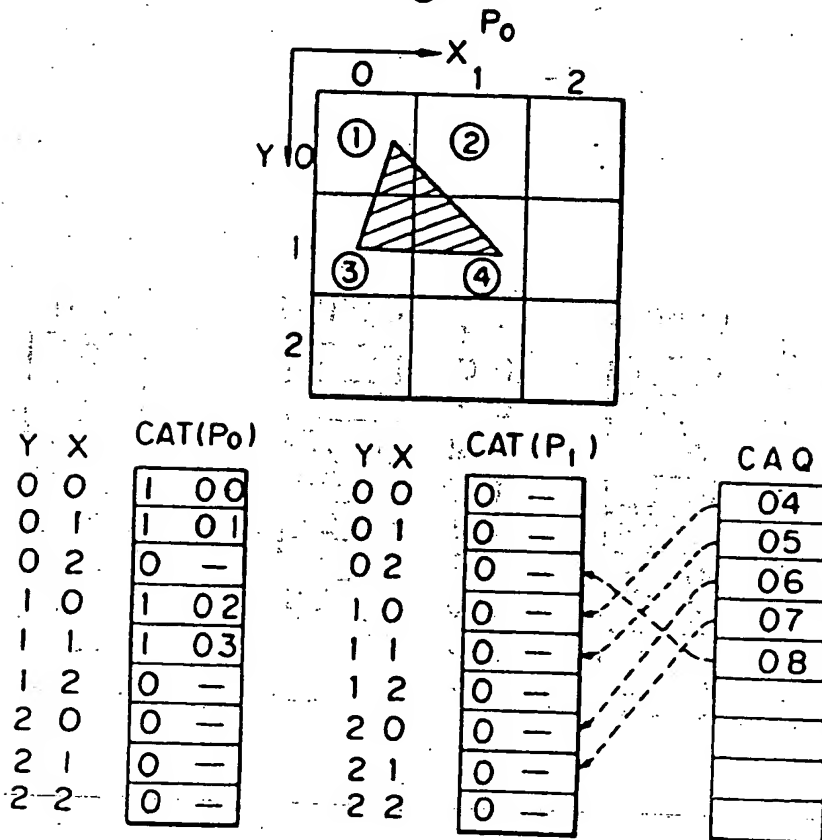


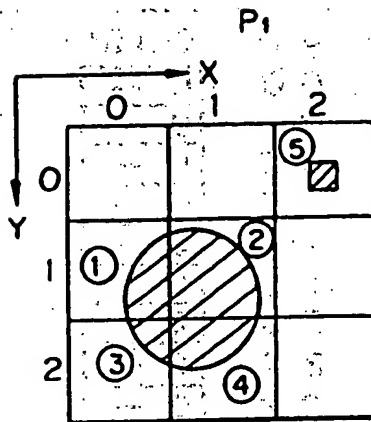
Fig. 7B





6/22

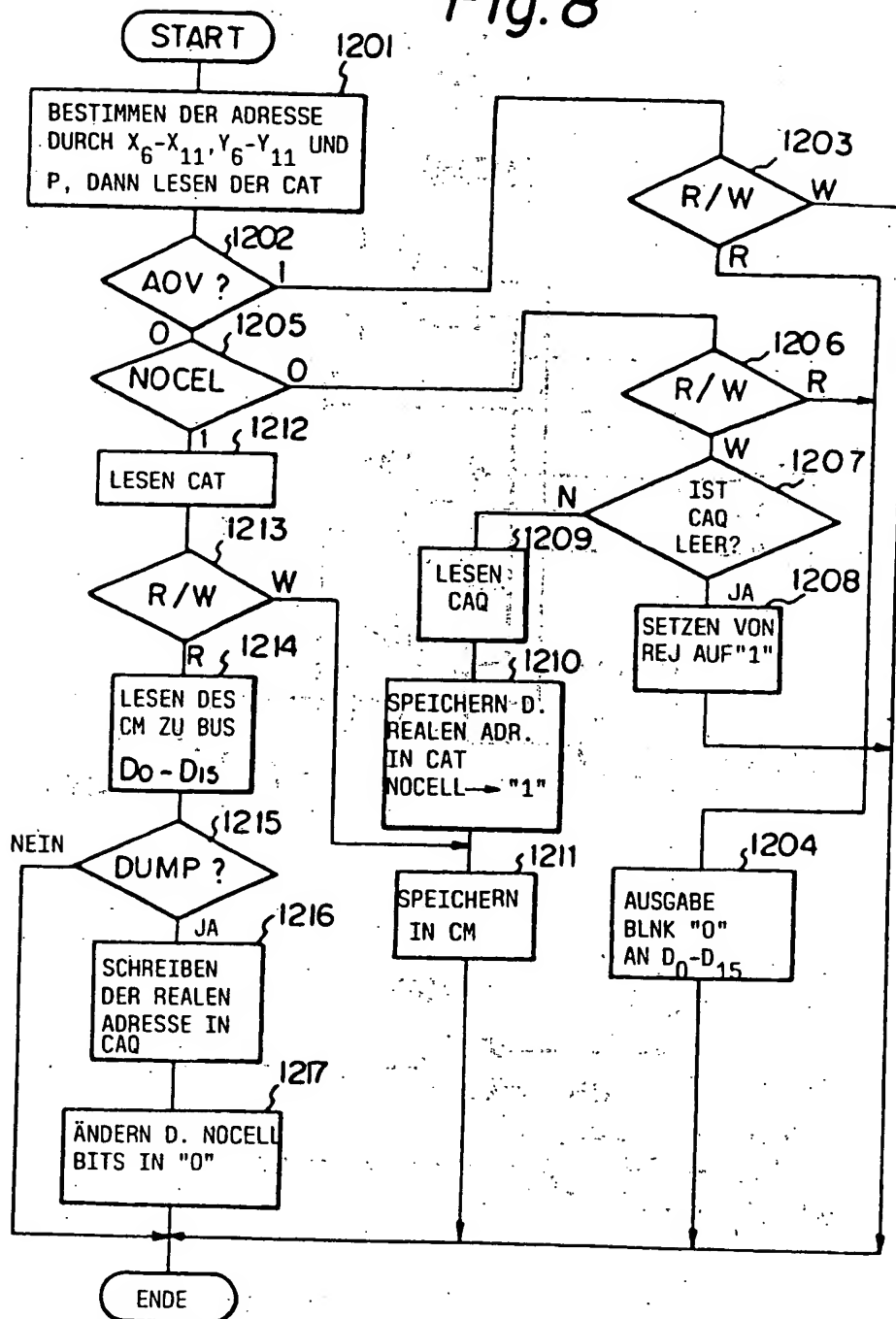
Fig. 7C



Y	X	CAT( $P_0$ )	Y	X	CAT( $P_1$ )	CA Q
0	0	0 00	0	0	0 —	00
0	1	0 01	0	1	0 —	01
0	2	0 —	0	2	1 08	02
1	0	0 02	1	0	1 04	03
1	1	0 03	1	1	1 05	
1	2	0 —	1	2	0 —	
2	0	0 —	2	0	1 06	
2	1	0 —	2	1	1 07	
2	2	0 —	2	2	0 —	

7/22

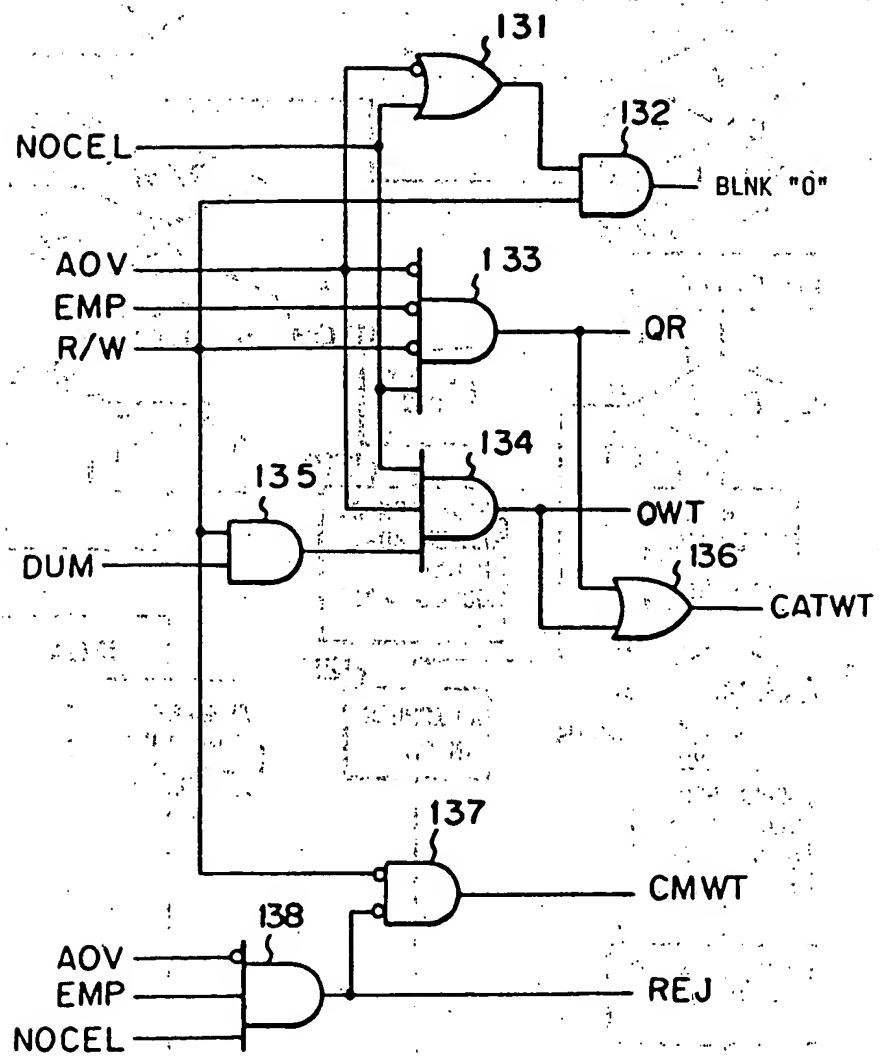
Fig. 8



8/22

Fig. 9

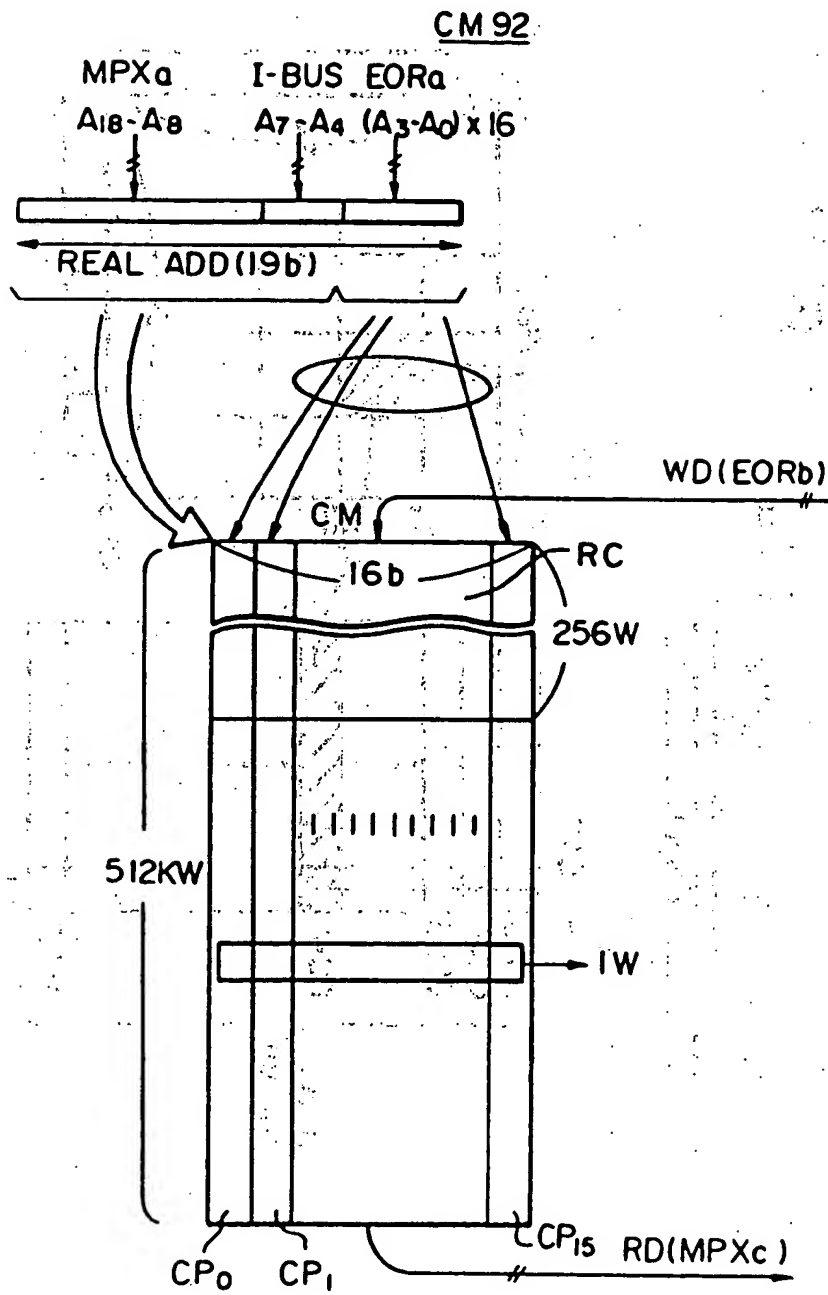
IMC 99





10/22

Fig. 11



11/22

Fig. 12

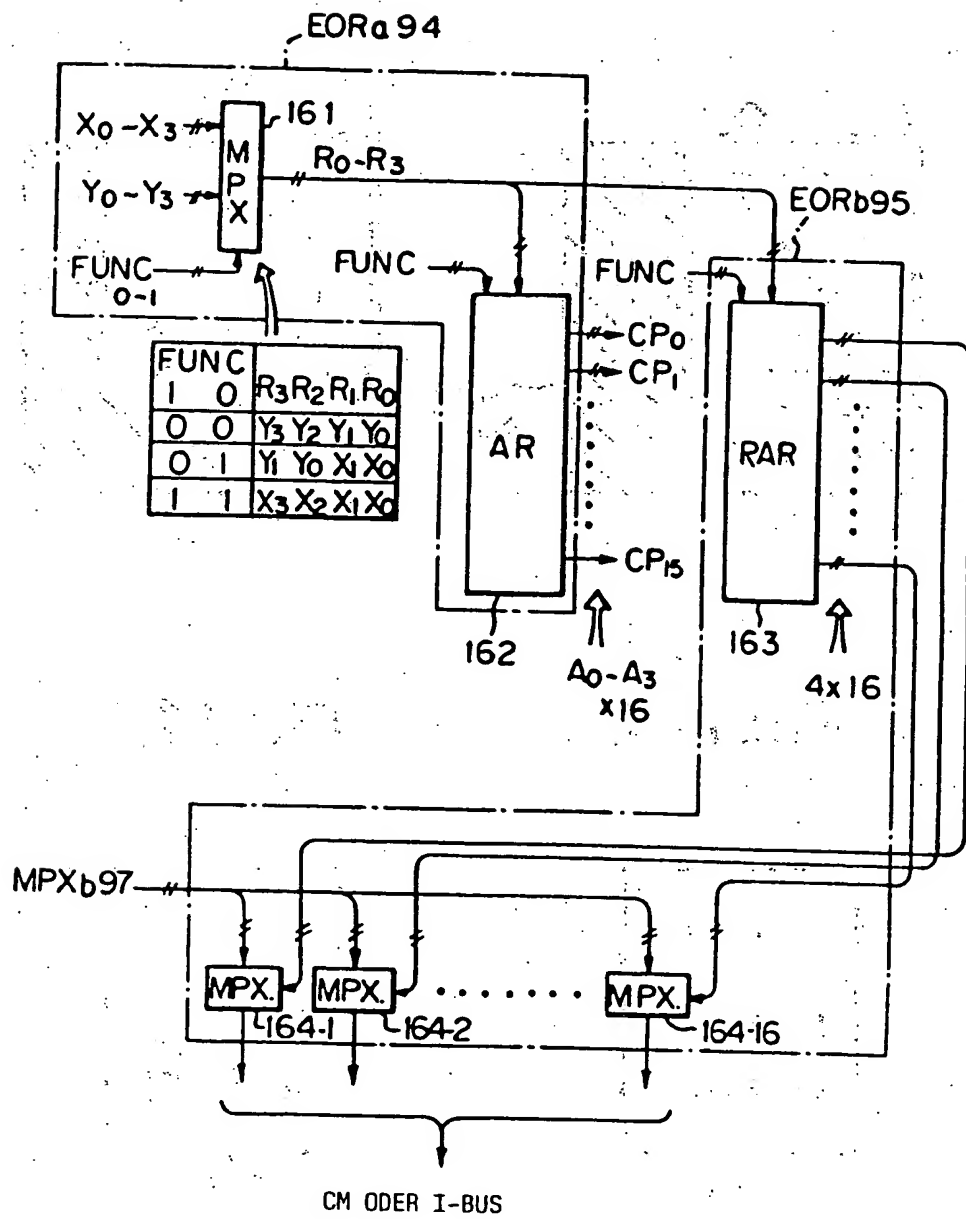


Fig. 13A

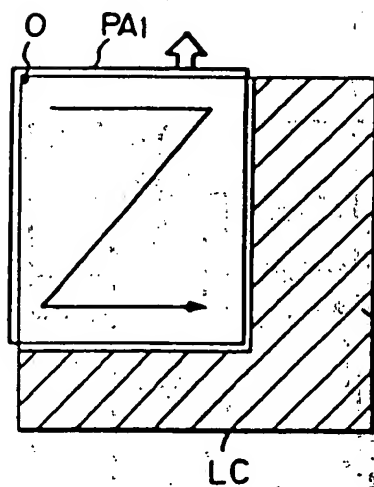


Fig. 13B

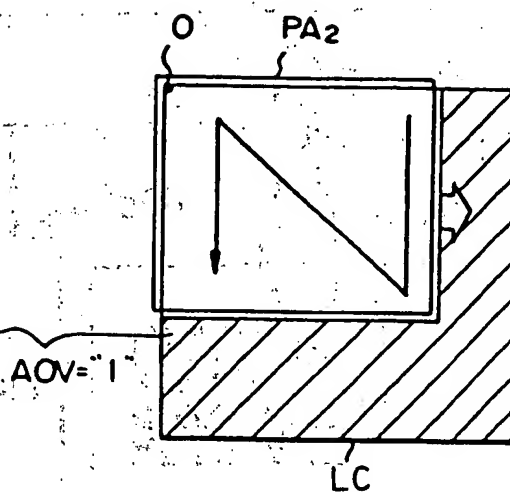


Fig. 14A

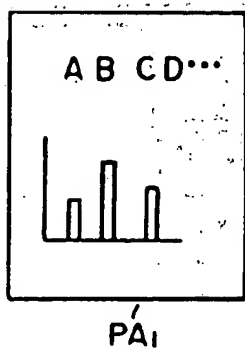
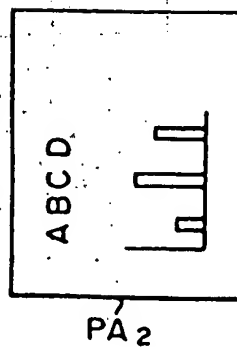


Fig. 14B



13/22

Fig.15 A

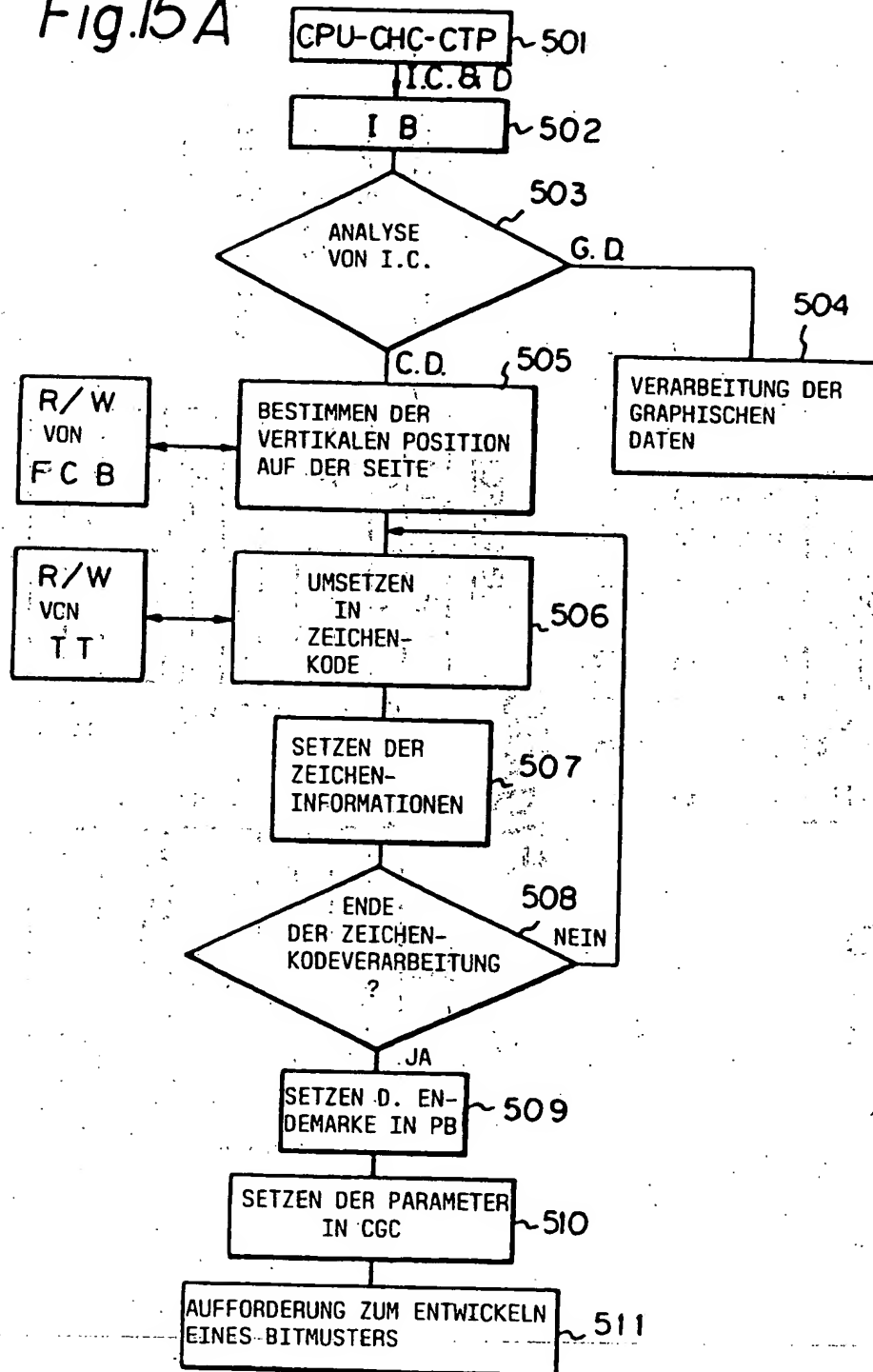




Fig.15 B

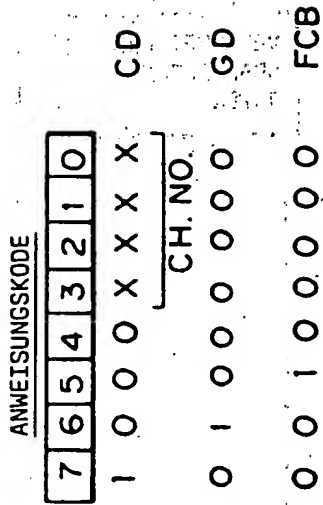


Fig.15C

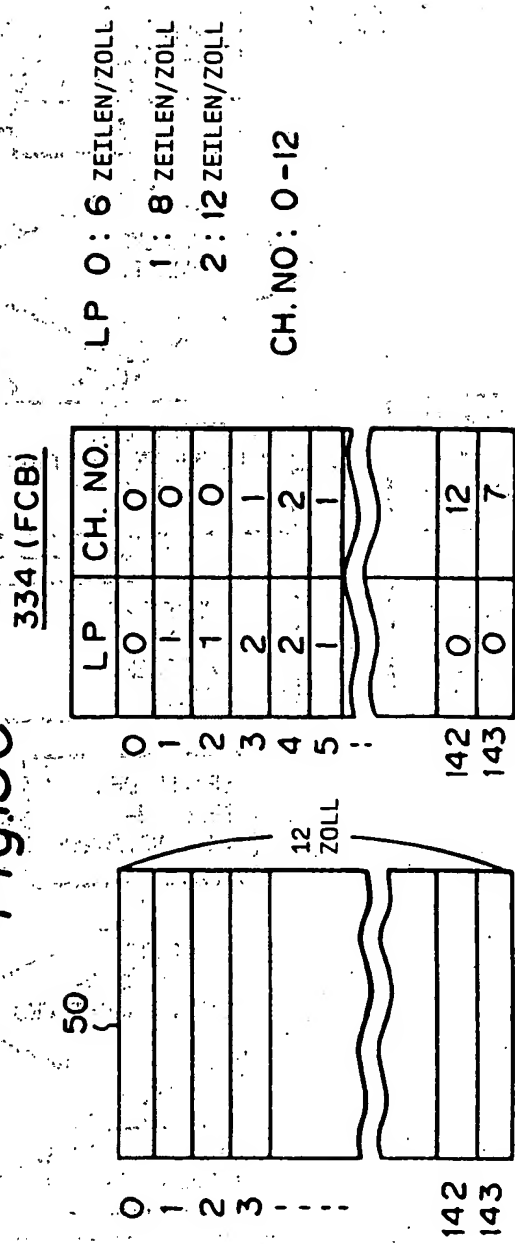


Fig.15D

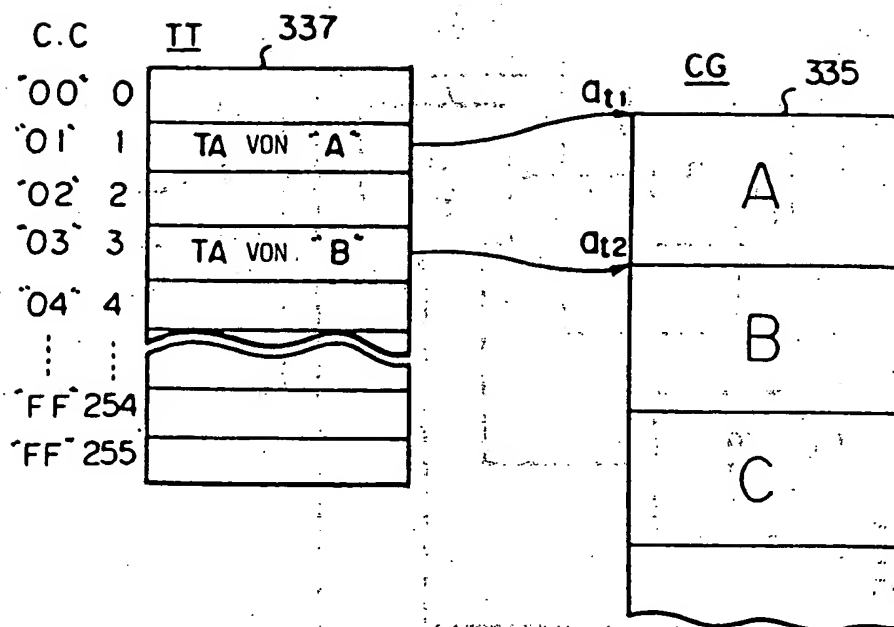


Fig.15E

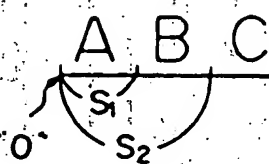
Diagram illustrating a data structure mapping between two tables, 331 and 335.

**Table 331 (Left):**

MA	TA	PDMA
0	$q_{k1}$	$q_{m1}$
$S_1$	$q_{k2}$	$q_{m2}$
$S_2$	$q_{k3}$	$q_{m3}$
...	...	...
EF		

Table 331 contains entries for  $q_{k1}$ ,  $q_{k2}$ , and  $q_{k3}$ . A wavy line indicates a continuation of the table.

Fig.15F



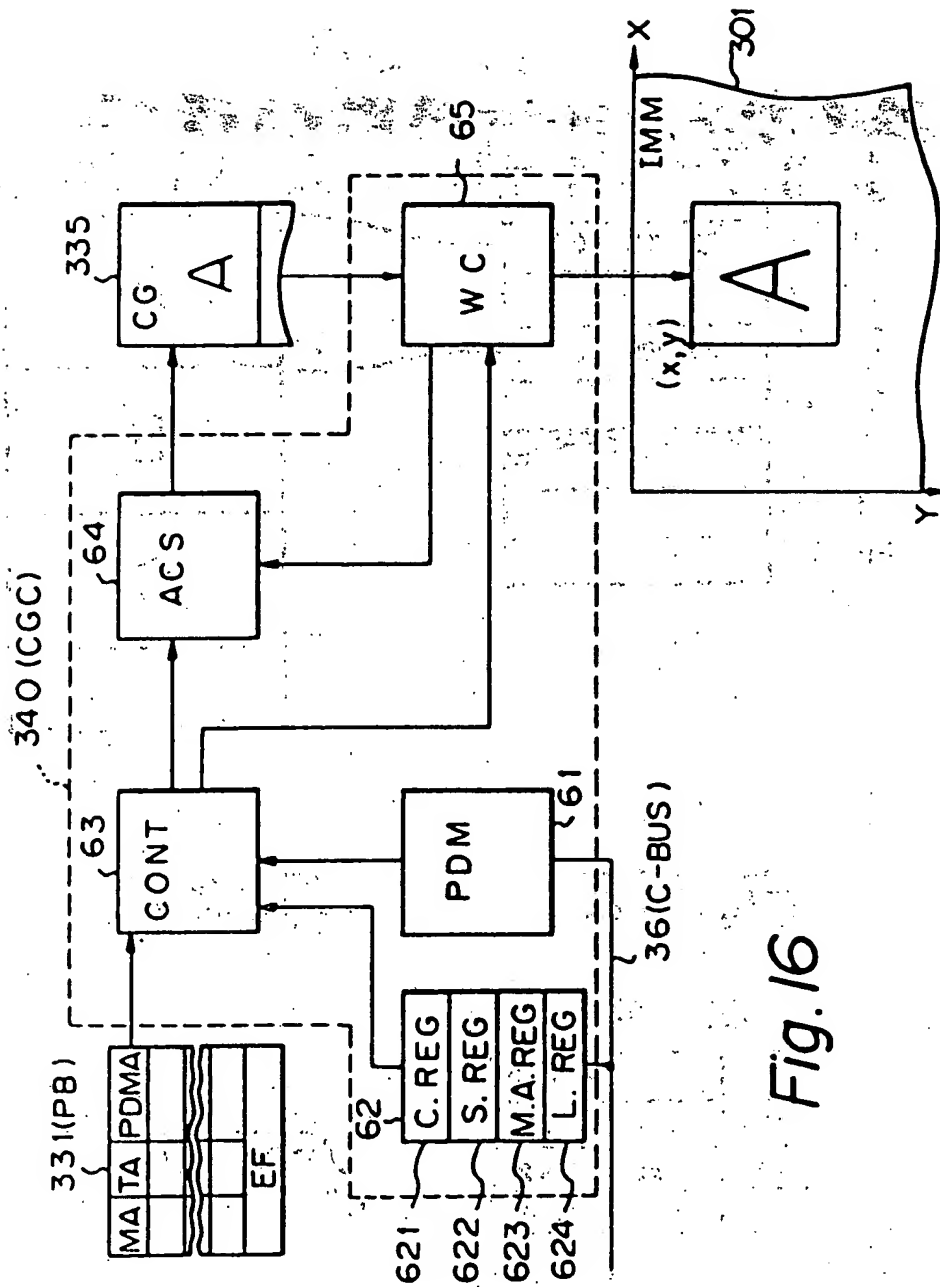


Fig. 16

Fig.17

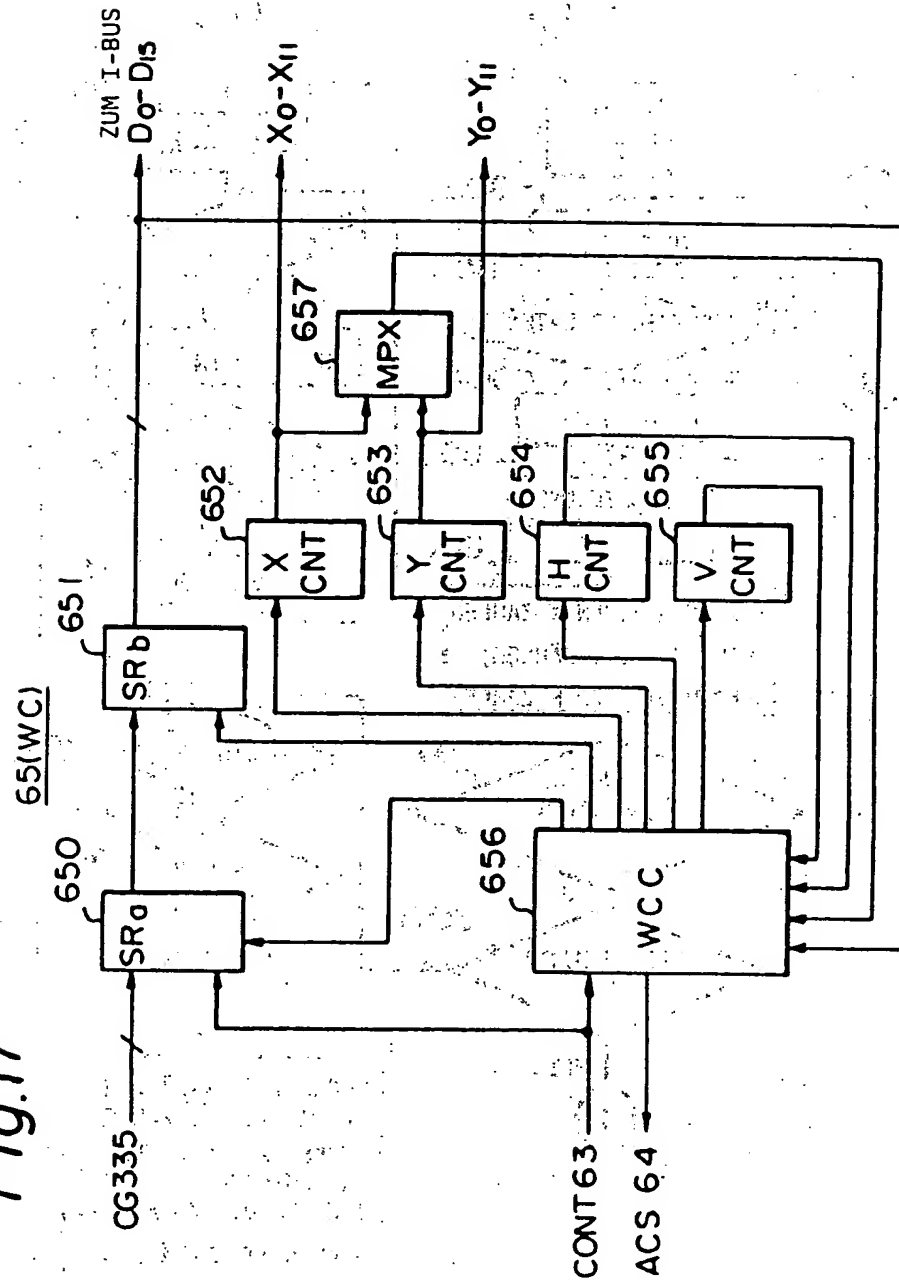


Fig. 18A

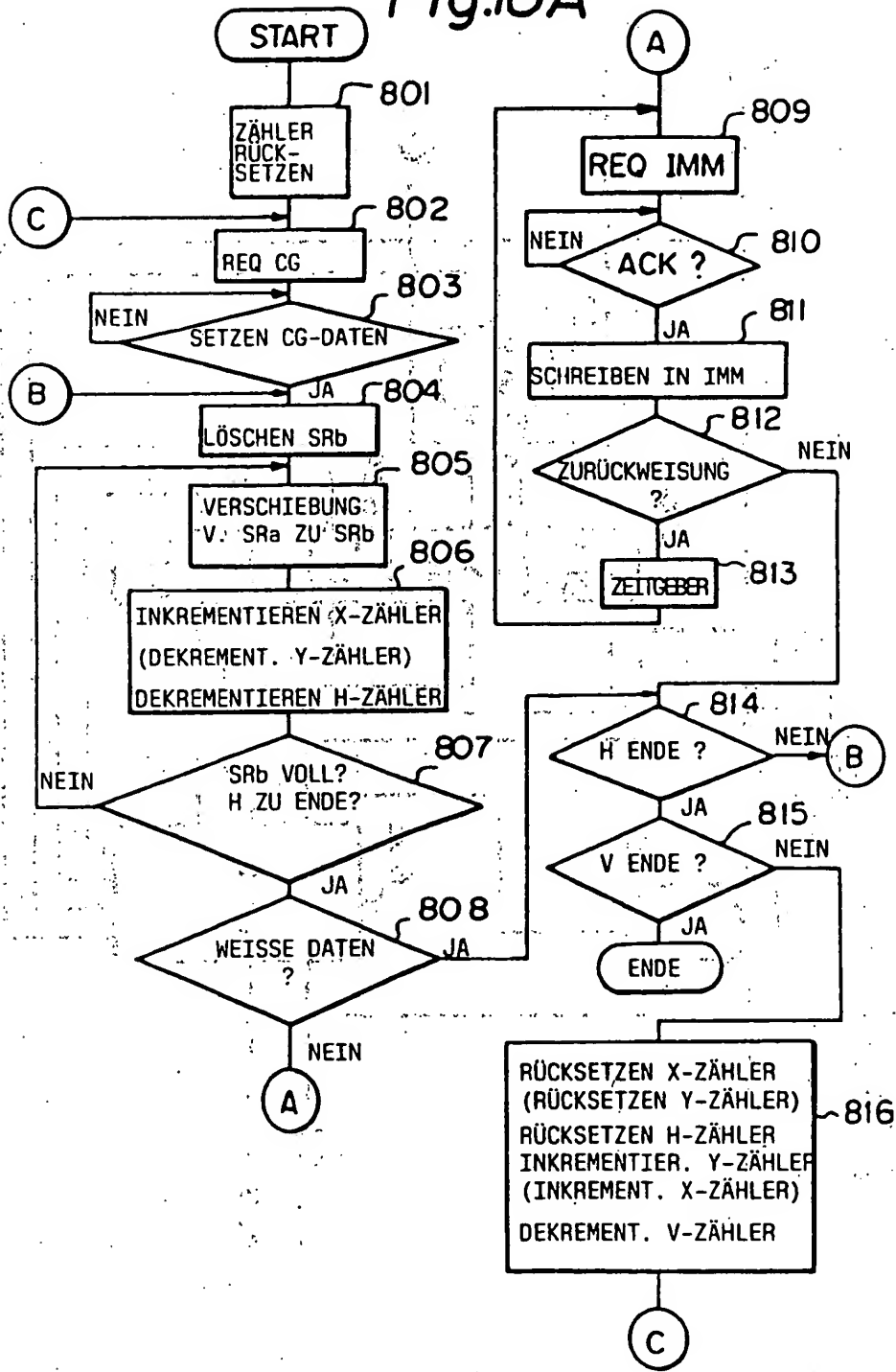




Fig. 19

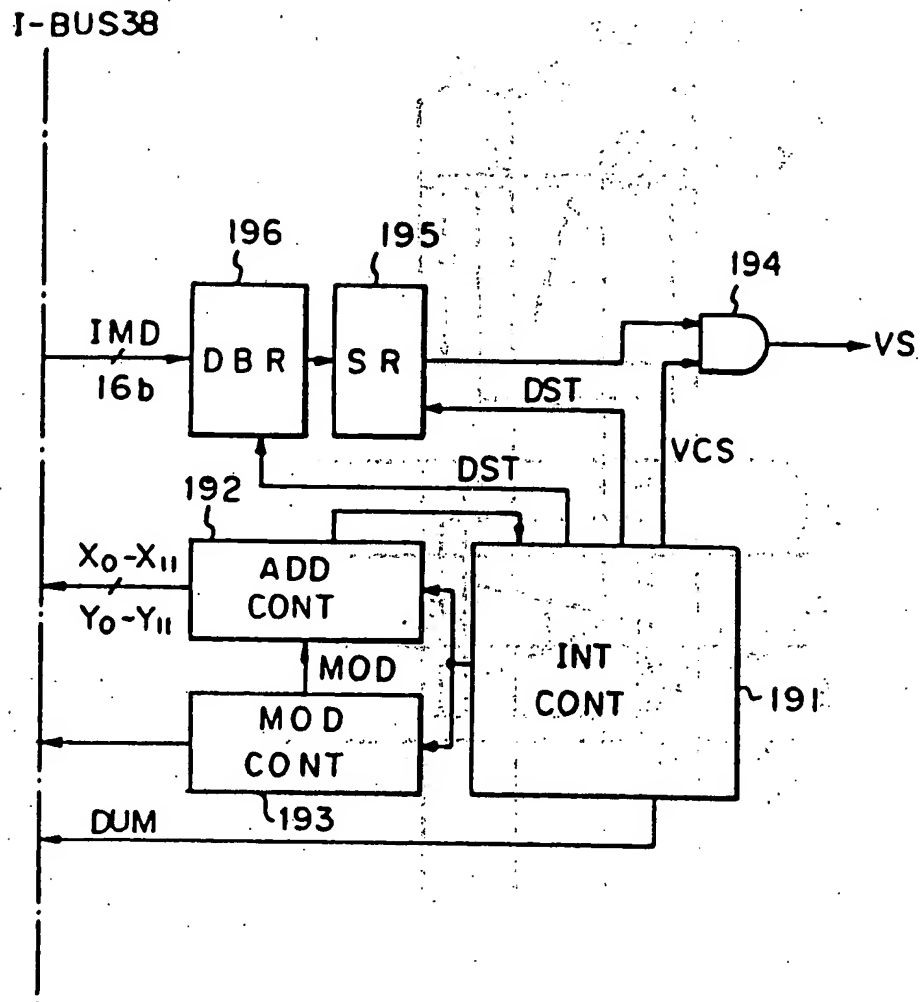


Fig. 20

INT CONT 191

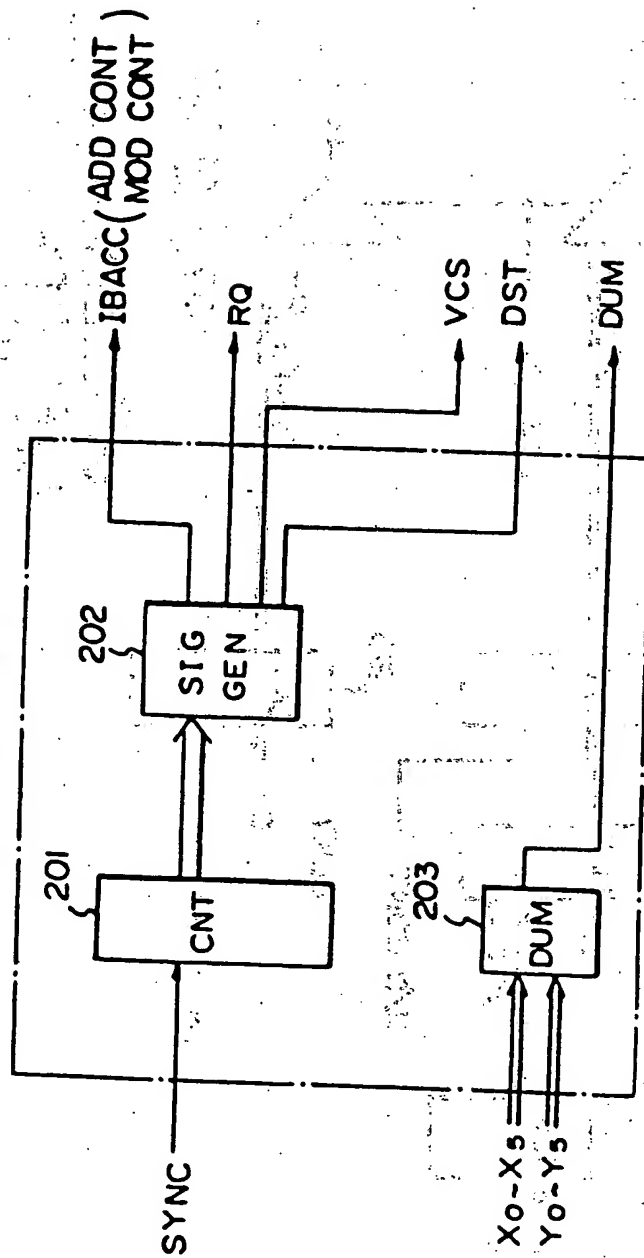




Fig. 21A

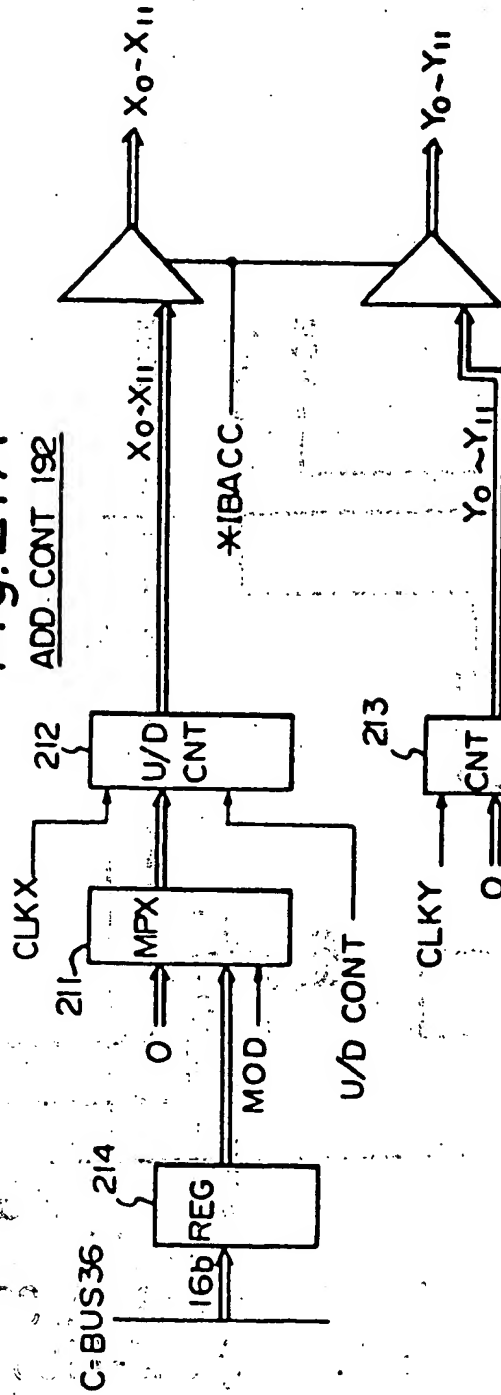
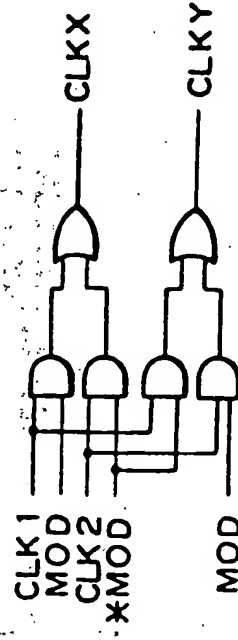


Fig. 21B



DOCKET NO: A-3172

SERIAL NO: 101075, 670

APPLICANT: Eisele

LERNER AND GREENBERG P.A.

P.O. BOX 2480

HOLLYWOOD, FLORIDA 33022

TEL. (954) 925-1100